

TUGAS AKHIR - TF145565

IMPLEMENTASI ALGORITMA KECERDASAN BUATAN MODIFIKASI PADA *DUAL-AXIS SOLAR TRACKING SYSTEM* DENGAN DESAIN *VERTICAL-AXIS ARRAY* BERBASIS STM32

REZZA MAULANA WAHID
105 1 15 000 00 048

Pembimbing 1 :
Dr.Ir. Purwadi Agus Darwito, M.Sc
NIP. 19620802 198803 1 001

Pembimbing 2 :
Murry Raditya, S.T, M.T
NPP. 1988 2017 11055

**PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya
2018**



FINAL PROJECT - TF145565

**IMPLEMENTATION OF MODIFIED ARTIFICIAL
INTELLIGENCE ALGORITHM ON DUAL-AXIS SOLAR
TRACKING SYSTEM WITH VERTICAL-AXIS ARRAY
DESIGN BASED ON STM32**

REZZA MAULANA WAHID
105 1 15 000 00 048

Supervisor 1 :
Dr.Ir. Purwadi Agus Darwito, M.Sc
NIP. 19620802 198803 1 001

Supervisor 2 :
Murry Raditya, S.T, M.T
NPP. 1988 2017 11055

STUDY PROGRAM DIII INSTRUMENTATION ENGINEERING
INSTRUMENTATION ENGINEERING DEPARTMENT
Faculty of Vocational
Sepuluh Nopember Institute of Technology
Surabaya
2018

LEMBAR PENGESAHAN I

"IMPLEMENTASI ALGORITMA KECERDASAN BUATAN MODIFIKASI PADA *DUAL-AXIS SOLAR TRACKING SYSTEM* DENGAN DESAIN *VERTICAL- AXIS ARRAY* BERBASIS STM32"

TUGAS AKHIR

Oleh :

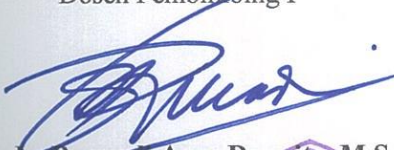
REZZA MAULANA WAHID

NRP. 105 1 15 000 00 048

Surabaya, 27 Juli 2018

Mengetahui / Menyetujui

Dosen Pembimbing I



Dr. Ir. Purwadi Agus Darwito, M.Sc

NIP. 19620822 198803 1 001

Dosen Pembimbing II



Murry Raditya, S.T, M.T

NPP. 1998 2017 11055

Kepala Departemen
Teknik Instrumentasi FV-ITS



Dr. Ir. Purwadi Agus Darwito, MSc

NIP. 19620822 198803 1 001

LEMBAR PENGESAHAN II

“IMPLEMENTASI ALGORITMA KECERDASAN BUATAN MODIFIKASI PADA *DUAL-AXIS SOLAR TRACKING SYSTEM* DENGAN DESAIN *VERTICAL- AXIS ARRAY* BERBASIS STM32”




TUGAS AKHIR

Diajukan Untuk Memperoleh salah satu syarat
Untuk memperoleh gelar Ahli Madya
Pada
Program Studi DIII Teknologi Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh :

**REZZA MAULANA WAHID
NRP.105 1 15 000 00 048**

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. Purwadi Agus Darwito, M.Sc  Pembimbing I
2. Murry Raditya, S.T, M.T.  Pembimbing II
3. Ir. Tutug Dhanardono, M.T.  Penguji I

**SURABAYA
2018**

IMPLEMENTASI ALGORITMA KECERDASAN BUATAN MODIFIKASI PADA *DUAL-AXIS SOLAR TRACKING SYSTEM* DENGAN DESAIN *VERTICAL- AXIS ARRAY* BERBASIS STM32

Nama : Rezza Maulana Wahid
NRP : 10511500000048
Departemen : Teknik Instrumentasi FV-ITS

Abstrak

Pemanfaatan energi surya sebagai pembangkit tenaga listrik sudah berkembang pesat. Perkembangan tersebut meliputi efektivitas konversi sel, desain peletakan sel surya, hingga algoritma tracking yang digunakan. Dewasa ini perkembangan berpusat pada algoritma tracking dengan menggunakan kecerdasan buatan, meliputi Fuzzy, Neural Network, dan ANFIS. Pada Penelitian ini, sebuah *dual-axis solar tracker* yang khusus dirancang untuk desain *vertical-axis array* telah berhasil dirancang dengan menggunakan algoritma ANFIS yang dimodifikasi pada mikrokontroler STM32. Label modifikasi digunakan pada algoritma ini karena proses keseluruhannya tidak dijalankan secara otomatis layaknya algoritma kecerdasan buatan, namun masih terdapatnya proses secara manual dalam pengerjaan algoritmanya.

Kata kunci: Solar Tracker, ANFIS, STM32

***IMPLEMENTATION OF MODIFIED ARTIFICIAL
INTELLIGENCE ALGORITHM ON DUAL-AXIS SOLAR
TRACKING SYSTEM WITH VERTICAL-AXIS ARRAY
DESIGN BASED ON STM32***

Name : Rezza Maulana Wahid
NRP : 10511500000048
Department : *Instrumentation Engineering FV-ITS*

Abstract

Utilization of solar energy as a power plant is growing rapidly. The developments include the effectiveness of cell conversion, the design of the laying of solar cells, to the tracking algorithm is used. Nowadays, the development is centered on tracking algorithms using artificial intelligence, including Fuzzy, Neural Networks and ANFIS. In this study, a dual-axis solar tracker specifically designed for the vertical-axis array design has been successfully designed using the modified ANFIS algorithm on the STM32 microcontroller. Label modifications are used in this algorithm because the whole process does not run automatically like an artificial intelligence algorithm, but there is still a manual process in its algorithm work.

Keywords : *Solar Tracker, ANFIS, STM32*

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas berkat, rahmat dan kebesaran-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul “Impelementasi Algoritma Kecerdasan Buatan Modifikasi Pada *Dual-Axis Solar Tracking System* Dengan Desain *Vertical-Axis Array* Berbasis STM32”

Selama menyelesaikan Tugas Akhir ini penulis telah banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc selaku kepala Departemen D3 Teknik Instrumentasi ITS, Surabaya. yang telah sabar memberikan dukungan, bimbingan, serta ilmu yang sangat bermanfaat.
2. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc selaku dosen pembimbing pertama Tugas Akhir yang senantiasa memberikan bimbingan dan arahan dalam menyelesaikan Tugas Akhir ini.
3. Bapak Murry Raditya, S.T., M.T. selaku dosen pembimbing kedua Tugas Akhir yang juga senantiasa selalu memberikan bimbingan dan arahan dalam menyelesaikan Tugas Akhir ini
4. Keluarga tercinta, Ibu Nunuk dan Ayah Wahid tercinta hingga Adik Tiara Tersayang, juga termasuk kerabat penulis ucapkan terima kasih atas dorongan semangatnya, bantuan dan dukungan serta doa-doa nya selama ini sehingga Tugas Akhir ini dapat selesai dengan baik.
5. Rekan-rekan team tugas akhir *Smart Classroom* atas kekompakan dan kerjasamanya.
6. Teman-teman D3 Teknik Instrumentasi angkatan 2015 yang membantu dan mensupport selama kegiatan Tugas Akhir berlangsung.
7. Teman seperjuangan dari satu kelompok praktikum TIK, Kerja Praktik, hingga Tugas akhir ini, Kevin dan Ergi yang

senantiasa saling membackup untuk masalah apapun terutama masalah finansial

8. Ayu Safitri selaku calon ehem-ehem, atas perhatian dan semangat kecilnya juga cintanya sehingga pengerjaan Tugas Akhir ini lebih berarti.
9. Warkop Kelopo yang selalu memberikan tempat merenung dan kopi untuk bermeditasi
10. Seluruh karyawan dan staff Departemen Teknik Instrumentasi yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa kesempurnaan hanya milik Allah SWT. Oleh karena itu, penulis sangat berterimakasih atas segala masukan, kritik dan saran yang membangun dari pembaca agar laporan ini menjadi lebih baik lagi untuk di kemudian hari. Demikian laporan ini penulis buat, semoga laporan ini dapat memberikan manfaat selain bagi penulis sendiri, dan bagi pembaca sekalian.

Surabaya, 27 Juli 2018

Rezza Maulana Wahid
NRP. 10 5115 00000 048

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN I	iii
LEMBAR PENGESAHAN II	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	viii
DAFTAR ISI	ixi
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah.....	3
1.5 Manfaat	4
1.6 Sistematika Laporan.....	4
BAB II TINJAUAN PUSTAKA	7
2.1 Fotovoltaik	7
2.2 <i>Solar Tracking System</i>	9
2.3 STM32.....	10
2.4 Algoritma Kecerdasan Buatan	16
2.5 Motor Stepper.....	19
2.6 Sensor LDR (<i>Light Dependent Resistor</i>)	22
2.7 Mbed OS	23
BAB III METODOLOGI PENELITIAN	25
3.1 Flowchart Penelitian.....	25

3.2 Perancangan Sistem	29
3.2.1 Perancangan Algoritma ANFIS	29
3.2.2 Realisasi Algoritma ANFIS	44
3.3 Eksperimen Algoritma ANFIS	52
BAB IV HASIL DAN PEMBAHASAN.....	53
4.1 Hasil Perancangan Algoritma ANFIS	523
4.2 Penjelasan <i>Listing Code</i>	524
4.3 Hasil Eksperimen Algoritma ANFIS	652
BAB V KESIMPULAN DAN SARAN	69
5.1 Kesimpulan	69
5.2 Saran	69
DAFTAR PUSTAKA	73
LAMPIRAN A (Data Z Fuzzy(Tk), Beban(Vij dan Vjk) dan Bias(Voj dan Wok) Neural Network, Error(E) ANFIS, Final Output ANFIS(Yk))	75
LAMPIRAN B (Listing Program of Modified ANFIS Algorithm)	89
LAMPIRAN C (Datasheet Nucleo STM32F401RE)	113
LAMPIRAN D (Datasheet Sensor LDR)	117
BIODATA PENULIS	123

DAFTAR GAMBAR

Gambar 2. 1	Kurva I-V pada modul PV Kyocera KC200GT: (a) Pada beberapa level radiasi, (b) Pada beberapa suhu sel	7
Gambar 2. 2	Rangkaian sel dioda model satu dioda	8
Gambar 2. 3	<i>Development Board STM32F401RE</i>	11
Gambar 2. 4	<i>Pinout</i> fitur untuk ekstensi arduino	14
Gambar 2. 5	<i>Pinout</i> fitur untuk ekstensi ST-Morpho	15
Gambar 2. 6	Sistem inferensi fuzzy Takagi-Sugeno-Kang	17
Gambar 2. 7	Arsitektur ANFIS	18
Gambar 2. 8	Bentuk Motor Stepper	21
Gambar 2. 9	Bentuk Fisik dan Simbol Sensor LDR	22
Gambar 3. 1	Diagram Blok Keseluruhan Sistem <i>Dual-Axis Solar Tracker</i>	25
Gambar 3. 2	<i>Flowchart</i> Penelitian	27
Gambar 3. 3	(a) Desain Tampak Depan(kiri) (b) Tampak Samping(kanan)	30
Gambar 3. 4	Desain Algoritma Fuzzy	31
Gambar 3. 5	(a) Derajat Keanggotaan Variabel Input(atas) (b) Derajat Keanggotaan Variabel Output(bawah) ...	32
Gambar 3. 6	<i>Plotting Inference Rule</i> pada <i>Rule Editor</i>	35
Gambar 3. 7	Nilai Input dan Output pada <i>Rule Viewer</i>	35
Gambar 3. 8	Input dan Output pada <i>Surface Viewer</i>	36
Gambar 3. 9	Desain Struktur ANFIS.....	39
Gambar 3. 10	Diagram Blok Fungsi Algoritma ANFIS	44
Gambar 3. 11	<i>Flowchart</i> LDR pada ADC	45
Gambar 3. 12	<i>Flowchart thread</i> NN	47
Gambar 3. 13	<i>Flowchart thread</i> Inferensi Rule	49
Gambar 3. 14	<i>Flowchart thread</i> Defuzifikasir	50
Gambar 3. 15	Desain HMI khusus <i>Smart ClassRoom</i>	51

DAFTAR TABEL

Tabel 3. 1 Variabel, Himpunan, Range, Parameter, Fungsi Keanggotaan Fuzzy	33
Tabel 3. 2 Inference <i>Rule</i> Fuzzy	34
Tabel 3. 3 Sampel data Output Fuzzy(Z)	36
Tabel 4. 1 Daftar Bobot sambungan tiap unit dan Bobot bias	53
Tabel 4. 2 Hasil Eksperimen 1: <i>Flaslight</i> pada LDR1	64
Tabel 4. 3 Hasil Eksperimen 2: <i>Flashlight</i> antara LDR1 dan LDR2	65
Tabel 4. 4 Hasil Eksperimen 3: <i>Flaslight</i> pada LDR2	65

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia dalam ulasan IRENA Remap 2017 memiliki sektor yang paling berperan dalam menghasilkan energi listrik terbanyak yaitu dari *solar cell* (sel surya). Sel Surya atau *Solar Cell* adalah suatu perangkat atau komponen yang dapat mengubah energi cahaya matahari menjadi energi listrik dengan menggunakan prinsip efek Photovoltaic [1][2]. Yang dimaksud dengan Efek Photovoltaic adalah suatu fenomena dimana munculnya tegangan listrik karena adanya hubungan atau kontak dua elektroda yang dihubungkan dengan sistem padatan atau cairan saat mendapatkan energi cahaya [3]. Hal inilah yang membuat sel surya juga disebut sebagai *sel PhotoVoltaic* (PV) [4][5].

Dalam perkembangannya, pemaksimalan sel surya telah sangat berkembang. Namun masih terdapat kendala utama pada sel PV yaitu efisiensi konversi [6]. Kendala tersebut biasanya disebabkan oleh pemasangan sel surya secara permanen dengan sudut elevasi yang tetap, sehingga sel surya tidak dapat selalu tegak lurus terhadap cahaya matahari dan menyebabkan sel surya kurang optimal dalam pengkonversian daya. Maka dari itu, perlunya suatu divais yang dapat mengarahkan permukaan sel surya agar selalu tegak lurus mengikuti gerak arah matahari sehingga pengkonversian daya lebih optimal. Divais yang digunakan untuk mengarahkan panel surya tersebut dinamakan *solar tracker* [7]. Sistem Solar Tracker terbagi beberapa jenis dan bisa diklasifikasikan berdasarkan beberapa kriteria [8]. Salah satu jenis *solar tracker* berdasar sumbu putarnya ialah *dual-axis solar tracker*. Yang mana jenis ini dapat bergerak pada dua sumbu sehingga dapat lebih bebas. Dengan *dual-axis* ini pun permasalahan gerak semu matahari juga dapat teratasi. Dalam

tugas akhir ini perancangan solar tracker merupakan pengembangan dari desain *horizontal-axis array* pada [3] dan [9] yang dikembangkan menjadi *vertical-axis array* sehingga dapat lebih unggul dalam efisiensi tempat.

Desain tersebutlah yang akan menjadi latar belakang tugas akhir ini, dimana *dual axis-solar tarcker* akan diintegrasikan pada desain *vertical-axis array* yang memanfaatkan sensor LDR (*Light Dependent Resistor*) untuk *men-sensing* posisi gerak matahari. Untuk menunjang agar *solar tracker* dapat saling bekerja sama dengan aktuator maka dibutuhkan sebuah algoritma yang mampu mengerjakan metode yang terbilang kompleks tersebut dengan respon yang cepat dan dapat mencegah pergerakan yang tidak perlu, sehingga titik dimana sel PV dapat terpapar matahari secara optimal dapat ditemukan [10]. Algoritma yang biasa dipakai untuk kebutuhan kompleks tersebut ialah algoritma kecerdasan buatan, seperti Fuzzy, PID, Neural Network, dan lain-lain [10]. Khusus pada penelitian ini akan digunakan algoritma ANFIS termodifikasi untuk mengimplementasikannya dibutuhkan perangkat prosesor digital yang mempunyai resource yang besar seperti mikroprosesor 32 bit keatas misalnya STM32 [10].

STM32 ialah Mikrokontroler berbasis inti prosesor 32 bit RISC ARM dari STMicrelectronics. Secara internal, masing-masing mikrokontroler ini terdiri dari inti prosesor, memori RAM statis, memori flash, antarmuka debugging, dan berbagai periferal. Mempunyai frekuensi clock tinggi, namun low power merupakan salah satu kelebihan mikrokontroler ini [11]. Sehingga STM32 ini merupakan pilihan mikrokontroler yang sangat cocok dalam implementasi pengendalian respon cepat [12]. Dengan beberapa ulasan diatas maka Tugas Akhir ini akan mengangkat judul “*Implementasi Algoritma Kecerdasan Buatan Modifikasi Pada Dual-Axis Solar Tracking System Dengan Desain Vertical-Axis Array Berbasis STM32*”. Dengan diimplementasinya algoritma ini,

maka diharapkan dapat tercapainya sebuah solar tracking system yang efisien sehingga nantinya dapat di aplikasikan pada *Smart Classroom* sebagai salah satu sumber tenaga.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka rumusan masalah dalam Tugas Akhir ini adalah :

1. Bagaimana mengimplementasikan algoritma kecerdasan buatan modifikasi pada *dual-axis solar tracking system* dengan desain *vertical-axis array* berbasis STM32 sebagai salah satu sumber energi pada *smart classroom* ?
2. Bagaimana meningkatkan daya keluaran sistem fotovoltaiik yang diperoleh *dual-axis solar tracking system* dengan desain *vertical-axis array* ?

1.3 Tujuan

Tujuan utama dari tugas akhir ini ialah untuk memberikan solusi pada rumusan masalah yaitu :

1. Menghasilkan divais yang mampu mengimplementasikan algoritma kecerdasan buatan modifikasi pada *dual-axis solar tracking system* dengan *desain vertical-axis array* berbasis STM32 sebagai salah satu sumber energi pada *smart classroom*.
2. Menghasilkan divais yang mampu memberikan peningkatan daya keluaran fotovoltaiik dengan *dual-axis solar tracking system* dengan desain *vertical-axis array*.

1.4 Batasan Masalah

Adapun yang menjadi batasan masalah pada Tugas Akhir ini ialah hanya implementasi algoritma kecerdasan buatan modifikasi pada *dual-axis solar tracker*, sehingga tidak termasuk:

1. Perancangan desain *vertical-axis array* pada *dual-axis solar tracking system*
2. Integrasi Sensor yang meliputi sensor LDR, sensor arus, dan sensor tegangan juga Aktuator berupa motor stepper pada *dual-axis solar tracking system*

1.5 Manfaat

Manfaat dari tugas akhir ini adalah untuk membantu memaksimalkan energi terbarukan yaitu energi dari panas matahari untuk dikonversi menjadi energi listrik dengan mengoptimalkan divais yang ada lalu memodifikasinya sesuai dengan kebutuhan yang direncanakan.

1.6 Sistematika Laporan

Adapun sistematika dari laporan ini yaitu terdiri 5 bab, dimana masing-masing bab tersusun dari beberapa sub-bab, yaitu:

1. BAB I PENDAHULUAN

Berisi tentang penjelasan latar belakang, perumusan masalah, tujuan, batasan masalah, dan sistematika laporan.

2. BAB II DASAR TEORI

Berisi tentang teori-teori penunjang tugas akhir meliputi fotovoltaiik, solar tracking system, STM32, algoritma kecerdasan buatan, motor stepper, sensor LDR, sensor arus dan tegangan.

3. BAB III METODOLOGI

Berisi tentang penjelasan secara detail langkah-langkah yang dilakukan untuk mencapai tujuan dari penelitian.

4. BAB IV ANALISA DATA DAN PEMBAHASAN

Pada bab ini merupakan tindak lanjut dari bab III, dimana jika telah dilaksanakan simulasi pengujian alat

sehingga memperoleh data, sehingga dapat dilakukan analisa data dan pembahasan.

5. BAB V KESIMPULAN DAN SARAN

Berisi tentang hasil yang didapatkan setelah dilakukan simulasi pengujian alat serta saran sebagai evaluasi terhadap sistem PV.

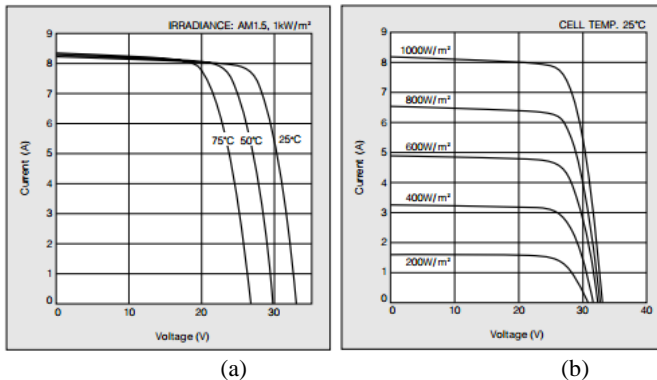
“Halaman ini sengaja dikosongkan”

BAB II

TINJAUAN PUSTAKA

2.1 Fotovoltaik

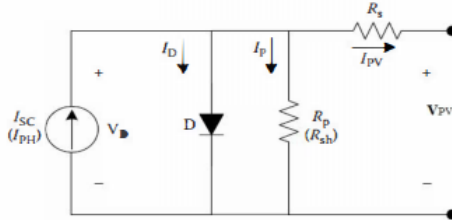
Fotovoltaik adalah suatu sistem yang bekerja dengan prinsip efek fotovoltaik yang mentransfer radiasi matahari menjadi energi listrik. Daya keluaran modul fotovoltaik bergantung pada dua faktor yaitu suhu sel dan radiasi matahari. Dua faktor tersebut tidak dapat dipisahkan antar satu dengan yang lain, karena faktor suhu sel dapat mempengaruhi tegangan keluaran modul PV, sedangkan radiasi matahari mempengaruhi arus keluaran modul PV. Setiap modul PV memiliki spesifikasi yang berbeda tergantung tipenya, sehingga karakteristik kurva I-V modul PV pun juga berbeda-beda. Pada Gambar 2.1 akan dijelaskan mengenai hubungan antara kurva I-V pada modul PV Kyocera KC200GT [13] :



Gambar 2.1 Kurva I-V pada modul PV Kyocera KC200GT: (a) Pada beberapa level radiasi, (b) Pada beberapa suhu sel

Untuk pemodelan sel surya, model satu dioda telah diusulkan, dan beberapa peneliti telah mempelajari cara mengekstrak

parameter untuk model. Pada Gambar 2.2 dijelaskan mengenai rangkaian sel dioda model satu dioda [13]:



Gambar 2.2 Rangkaian sel dioda model satu dioda

Berdasarkan rangkaian pada Gambar 2.2 dapat diketahui beberapa rumus berikut:

$$I_D = I_0 \left[e^{\frac{V_{PV}}{\alpha V_T}} - 1 \right] \quad (2.1)$$

$$I_{PV} = I_{SC} - I_0 \left(e^{\frac{q(V_{PV} + I_{PV} \cdot R_s)}{\alpha k T}} - 1 \right) - \frac{V_{PV} + I_{PV} \cdot R_s}{R_{sh}} \quad (2.2)$$

$$I_{PV} = I_{SC} - I_D \quad (2.3)$$

$$V_{PV} = \alpha V_T \ln \left[\frac{(I_{SC} - I_{PV})}{I_0} + 1 \right] \quad (2.4)$$

Dimana:

I_D = Arus Dioda

I_0 = Arus Saturasi Balik

V_{PV} = Tegangan Panel

V_T = Tegangan Suhu

I_{PV} = Arus yang melalui Panel

I_{SC} = Arus Foton

α = Faktor ideal PV

$$V_T = kT/q \quad (2.5)$$

K = Konstanta Boltzmann (J/K) (1.380×10^{-23})

T = Suhu sel PV (K)

Q = daya elektron (C) (-1.602×10^{-19})

2.2 *Solar Tracking System*

Solar Tracking System atau dalam bahasa Indonesia sistem penjejak matahari ialah sebuah alat atau divais yang dapat mengarahkan sel PV agar lebih fleksibel untuk terus mengikuti arah matahari, dibuat dengan tujuan untuk menaikkan output daya yang dihasilkan oleh sel PV. Komponen atau alat-alat yang diperlukan untuk membuat sistem ini ialah sel PV, aktuator berupa motor servo, driver motor, sensor matahari, catu daya, dan mikrokontroler atau alat pemroses digital. Dalam pembagiannya berdasarkan sumbu peletakan sel PV, *solar tracking system* dibagi menjadi 4 yaitu:

- *Single-axis tracking*

Sel PV diletakkan pada dua penyangga dalam keadaan pivot, penyangga bersifat statis, sedangkan untuk bergerak sel PV memanfaatkan aktuator yaitu poros yang dapat dikendalikan dengan motor untuk menggerakkannya pada satu sumbu saja mengikuti arah matahari yaitu sejak matahari terbit hingga tenggelam atau dari timur ke barat.

- *Dual-axis tracking*

Sel PV diletakkan pada satu penyangga dalam keadaan center, penyangga bersifat statis, sedangkan untuk bergerak sel PV memanfaatkan aktuator pada ujung-ujung penyangga yang sifatnya dinamis sehingga dapat dikendalikan dengan motor untuk menggerakkannya pada dua sumbu dan dapat mengikuti kemanapun arah matahari sejak matahari terbit hingga tenggelam, sehingga tidak terpaku dari timur ke barat.

- *Tilted single-axis tracking*

Hampir sama dengan *single-axis* namun salah satu penyangga dibiarkan tetap atau statis dan penyangga yang satunya dibuat menjadi dinamis. Sehingga sel PV akan bergerak dalam satu sumbu namun dengan sudut elevasi 30-45° mengikuti arah matahari.

- *Azimuth tracking*

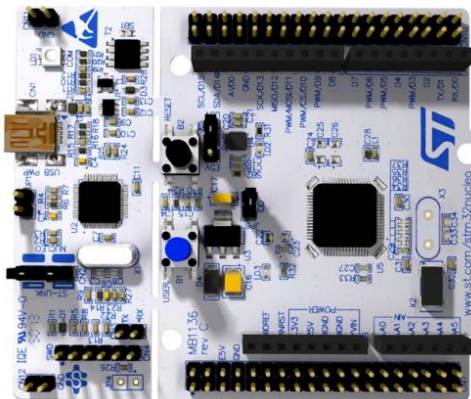
Hampir sama dengan *dual-axis* namun, ujung yang menempel sel PV dibiarkan tetap atau statis dalam sudut elevasi 30-45°. Sedangkan yang bergerak ialah bagian ujung bawahnya atau pangkalnya yang mana mengikuti arah matahari.

2.3 STM32

STM32 merupakan keluarga mikrokontroler 32 bit oleh STMicroelectronics yang melisensikan ARM prosesor dari *ARM Holdings*. ARM sendiri merupakan sebuah arsitektur prosesor 32-bit RISC yang dikembangkan oleh *ARM Limited* awalnya. Dikenal sebagai *Advanced RISC Machine* di mana sebelumnya dikenal sebagai *Acorn RISC Machine*. Pada awalnya merupakan prosesor desktop yang sekarang didominasi oleh keluarga x86. Namun desain yang sederhana membuat prosesor ARM cocok untuk aplikasi berdaya rendah. Hal ini membuat prosesor ARM mendominasi pasar *mobile electronic* dan *embedded system* di mana membutuhkan daya dan harga yang rendah. Pada arsitektur ARM, kata RISC merupakan kepanjangan dari *Reduced Instruction Set Computing* yang jika diterjemahkan berarti "Komputasi Kumpulan Instruksi yang Disederhanakan", merupakan sebuah desain dari arsitektur komputer atau arsitektur komputasi modern dengan instruksi-instruksi dan jenis eksekusi yang paling sederhana. Arsitektur ini digunakan pada komputer dengan kinerja tinggi, seperti komputer vektor. Selain digunakan

dalam komputer vektor, desain ini juga diimplementasikan pada prosesor komputer lain, seperti pada beberapa mikroprosesor Intel 960, Itanium (IA64) dari *Intel Corporation*, dan masih banyak lagi.

Chip STM32 dikelompokkan ke dalam rangkaian terkait yang berbasis di sekitar inti prosesor ARM 32-bit yang sama, seperti Cortex M7F hingga Cortex-M0. Secara internal, masing-masing mikrokontroler terdiri dari inti prosesor, memori RAM statis , memori flash , antarmuka debugging, dan berbagai periferal. Selain itu juga STM32 ini mempunyai frekuensi clock tinggi, yang umumnya berada pada kisaran 72MHz atau lebih, namun *low power* Sehingga STM32 ini merupakan pilihan mikrokontroler yang sangat cocok dalam implementasi pengendalian respon cepat. Pemrograman STM32 ini dapat dilakukan dengan menggunakan bahasa pemrograman *C-adaptive*. Salah satu contoh mikrokontroler dari STM32 ialah STM32F401RE atau yang sering disebut nukleo dalam bentuk *board*-nya. Pada Gambar 2.3 akan disajikan gambar *board* STM32F401RE ini.



Gambar 2.3 Development Board STM32F401RE

STM32 nukleo ialah board mikrokontroler yang dilengkapi dengan *chip* STM32F401RE. Board ini dibuat dengan desain

seminim mungkin sehingga mudah untuk dibawa kemanapun dan juga fleksibel dalam hal *programming* bagi pengguna untuk mencoba ide-ide baru dan membangun prototaip. Tersedianya dukungan konektivitas ke Arduino dan header ST-Morpho memudahkan untuk memperluas fungsionalitas platform pengembangan terbuka STM32 Nukleo ini dengan berbagai pilihan modul khusus. Walaupun board ini kecil, namun kaya akan fitur. Adapun fitur tersebut antara lain:

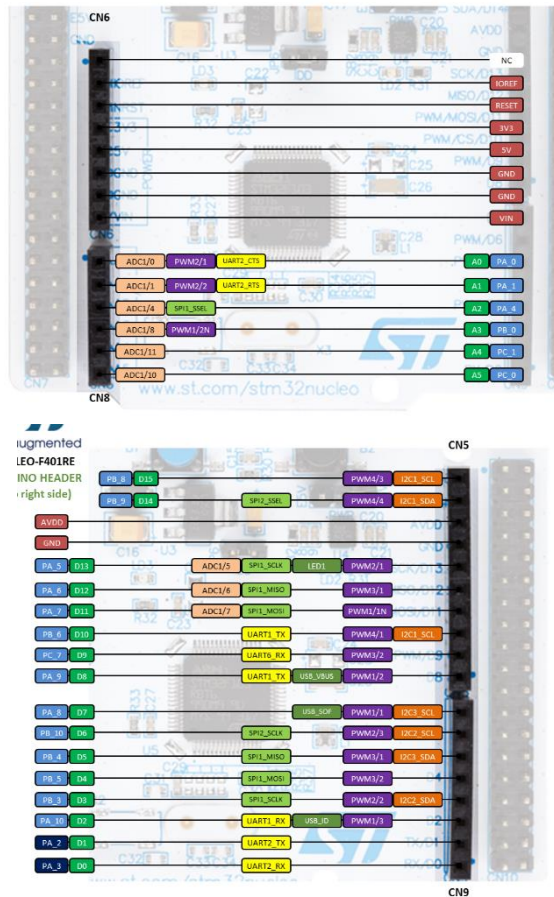
- Dua tipe sumber ekstensi
 - Arduino Uno R3
 - Ekstensi pin header STMicroelectronics Morpho untuk akses penuh ke semua I/O STM32
- *On-board ST-LINK/V2-1 debugger/programmer* dengan konektor SWD
 - switch untuk memilih menggunakan ST-LINK atau V2-1 debugger
- *Flexible board power supply*
 - USB VBUS atau sumber eksternal (3.3 V, 5 V, 7 - 12 V)
 - Jalur akses manajemen daya
- LED (LD2)
- Dua push buttons: USER and RESET
- *USB re-enumeration: three different interfaces supported on USB*
 - *Virtual Com port*
 - *Mass storage (USB Disk drive) untuk drag'n'drop programming*
 - *Debug port*

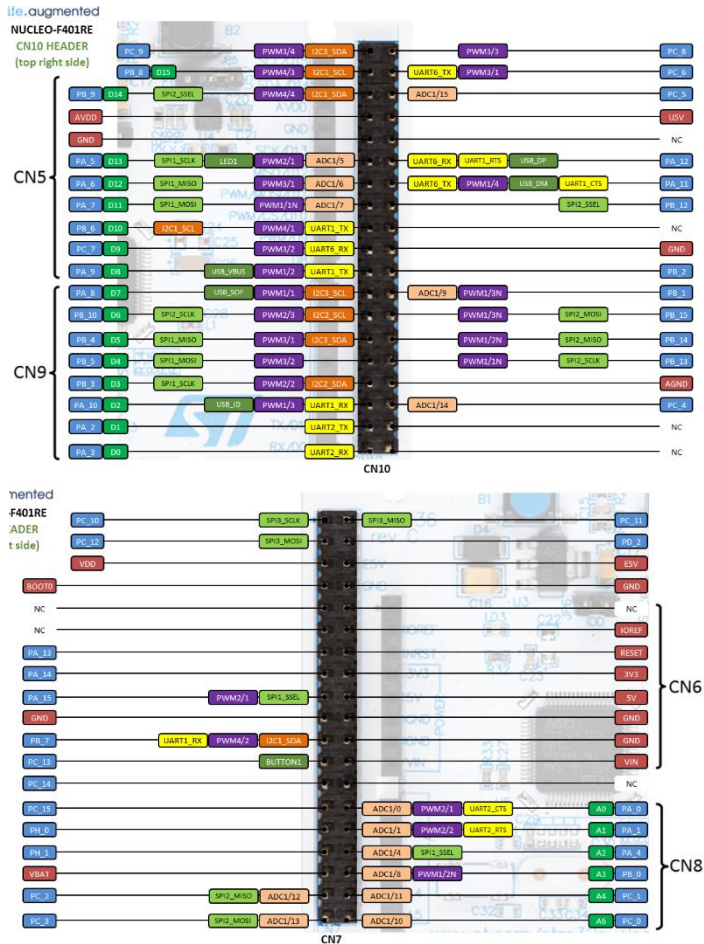
Sedangkan untuk chipnya sendiri memiliki fitur antara lain:

- STM32F401RET6 dalam paket LQFP64

- ARM®32-bit Cortex®-M4 CPU dengan FPU
- Frekuensi CPU maks. 84 MHz
- VDD dari 1.7 V ke 3.6 V
- 512 KB Flash
- 96 KB SRAM
- GPIO (50) dengan kemampuan interupsi eksternal
- ADC 12-bit dengan 16 saluran
- RTC
- Timer Kontrol Lanjutan
- *General Purpose Timers* (7)
- *Watchdog Timers* (2)
- USART / UART (4)
- I2C (3)
- SPI (3)
- SDIO
- USB 2.0 OTG FS

Dikarenakan terdapatnya dua jenis pin ekstensi pada board ini, maka tiap ekstensinya pun memiliki pengalamatan tertentu untuk tiap fitur yang disediakan. Pada Gambar 2.4 akan disajikan pinout dari ekstensi untuk arduino.





sumber: <https://os.mbed.com/platforms/ST-Nucleo-F401RE/>

Gambar 2.5 *Pinout* fitur untuk ekstensi ST-Morpho

Gambar-gambar pinout pada gambar 2.4 dan 2.5 menjelaskan mengenai pin yang harus digunakan apabila ingin menggunakan salah satu fitur pada *board* tersebut. Misalnya untuk menggunakan

fitur ADC, dapat digunakan pin PA_0, PA_1 pada ekstensi arduino maupun pada ekstensi STM-Morpho

2.4 Algoritma Kecerdasan Buatan

Algoritma diartikan sebagai kumpulan dari metode efektif yang diekspresikan dalam bentuk instruksi-instruksi definitif, bersifat sistematis dan logis yang dibuat untuk tujuan memecahkan sebuah permasalahan atau dapat mencapai tujuan tertentu. Sedangkan kecerdasan buatan merujuk pada arti kecerdasan yang dibuat oleh manusia untuk dimasukkan pada sebuah mesin atau sistem agar dapat bersifat lebih manusiawi yaitu dapat berkembang selayaknya manusia. Ada banyak kecerdasan buatan yang dibuat dengan tujuan tertentu atau untuk tujuan secara umum, seperti *Fuzzy Logic*, *Neural-Network*, *PID (Proportional-Integral-Derivative)*, dan masih banyak lagi. Dengan banyaknya algoritma kecerdasan buatan ini, membuat manusia berlomba dalam mengembangkan kecerdasan buatan tersebut dengan mengimplementasikannya ke segala bidang. Bahkan hingga tercipta algoritma kecerdasan buatan hibrid, yaitu penggabungan antara dua atau lebih algoritma kecerdasan buatan dengan atau tidak dengan modifikasi, misalnya ANFIS, yang dewasa ini cukup marak dikembangkan dalam berbagai sistem karena kecepatan proses dan responnya, walau butuh perancangan yang sangat kompleks namun dengan hasil yang dapat menaikkan output dari segala sisi inilah yang membuat algoritma ini cukup banyak dikembangkan.

ANFIS atau kepanjangan dari *Adaptive Neuro-Fuzzy Inference System* yang merupakan penggabungan mekanisme *Fuzzy Inference System (FIS)* yang digambarkan dalam arsitektur jaringan syaraf tiruan atau *Neural-Network*. Lalu untuk sistem inferensi fuzzy yang digunakan adalah sistem inferensi fuzzy

model Takagi-Sugeno-Kang (TSK). ANFIS adalah arsitektur yang secara fungsional sama dengan fuzzy rule base model Sugeno. Bisa dikatakan bahwa ANFIS adalah suatu metode yang mana dalam melakukan penyetelan aturan digunakan algoritma pembelajaran terhadap sekumpulan data. Pada ANFIS juga memungkinkan aturan-aturan untuk beradaptasi. Berikut ini salah satu contoh ilustrasi mekanisme inferensi fuzzy TSK orde satu dengan dua masukan x dan y seperti pada Gambar 2.6 [14]

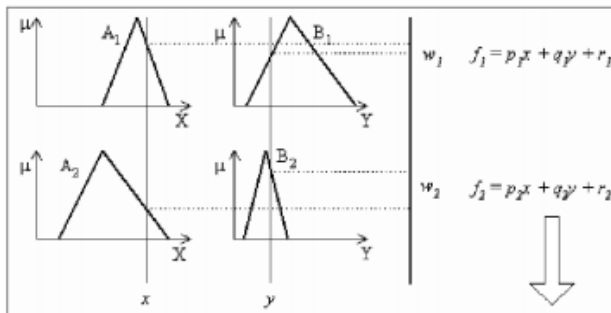
Basis aturan dengan dua aturan fuzzy if-then seperti dibawah ini :

Rule 1: if x is A_1 and y is B_1 (premis) then $\underline{f_1 = p_1x + q_1y + r_1}$ (consequent)

Rule 2: if x is A_2 and y is B_2 (premis) then $\underline{f_2 = p_2x + q_2y + r_2}$ (consequent)

Input: x and y

Consequent: f

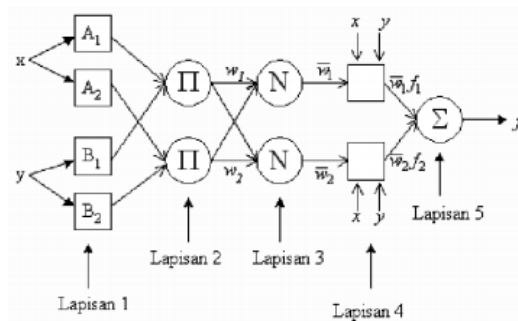


Gambar 2.6 Sistem inferensi fuzzy Takagi-Sugeno-Kang.

Keluaran fuzzy dapat dibuat dalam bentuk persamaan sebagai berikut:

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} = \bar{w}_1 f_1 + \bar{w}_2 f_2 \quad (2.6)$$

Struktur ANFIS yang menggambarkan sistem fuzzy Tagaki-Sugeno-Kang (TSK) seperti di atas bisa digambarkan dalam diagram blok atau disebut arsitektur jaringan syaraf *feedforward* seperti pada Gambar 2.7 [14].



Gambar 2.7 Arsitektur ANFIS

Pada Gambar 2.7 terlihat sistem *neuro-fuzzy* terdiri atas lima lapisan dengan fungsi yang berbeda untuk tiap lapisannya. Tiap lapisan terdiri atas beberapa simpul yang dilambangkan dengan kotak atau lingkaran. Lambang kotak menyatakan simpul adaptif artinya nilai parameternya bisa berubah dengan pembelajaran dan lambang lingkaran menyatakan simpul non adaptif yang nilainya tetap. Penjelasan lebih lengkap dapat dijabarkan sebagai berikut:

- Lapisan 1. Pada lapisan pertama ini, merupakan lapisan yang terdiri dari simpul-simpul adaptif atau nilai yang dapat berubah-ubah. Pada struktur ANFIS lapisan ini biasanya merupakan derajat keanggotaan (μ_i) dari fuzzy set.

- Lapisan 2. Pada lapisan ini, semua simpul adalah parameter-parameter tetap (non adaptif). Simpul ini berguna untuk mengalikan setiap input yang masuk.
- Lapisan 3. Simpul ini merupakan simpul non adaptif yang dapat menampilkan fungsi derajat pengaktifan ternormalisasi yaitu perbandingan antara keluaran simpul ke- i pada lapisan sebelumnya terhadap seluruh keluaran lapisan sebelumnya.
- Lapisan 4. Simpul pada lapisan ini adalah adaptif dengan derajat pengaktifan ternormalisasi yang berasal dari lapisan sebelumnya.
- Lapisan 5. Simpul non adaptif pada lapisan ini menjumlahkan seluruh keluaran dari simpul-simpul pada lapisan sebelumnya dan menghasilkan satu simpul keluaran

Kelima lapisan ini ekuivalen dengan sistem inferensi Fuzzy Takagi-Sugeno. Dengan ekivalensinya kelima lapisan diatas maka algoritma ANFIS dapat dimodifikasi.

2.5 Motor Stepper

Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkannya diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Motor stepper merupakan perangkat pengendali yang mengkonversikan bit-bit masukan menjadi posisi rotor. Bit-bit tersebut berasal dari terminal-terminal input yang ada pada motor stepper yang menjadi kutub-kutub magnet dalam motor. Bila salah satu terminal diberi sumber tegangan, terminal tersebut akan mengaktifkan kutub di dalam magnet sebagai kutub utara dan kutub yang tidak diberi tegangan sebagai kutub selatan. Dengan terdapatnya dua kutub di dalam motor ini, rotor di dalam motor yang memiliki kutub magnet permanen akan mengarah sesuai

dengan kutub-kutub input. Kutub utara rotor akan mengarah ke kutub selatan stator sedangkan kutub selatan rotor akan mengarah ke kutub utara stator.

Prinsip kerja motor stepper mirip dengan motor DC, sama-sama dicatu dengan tegangan DC untuk memperoleh medan magnet. Bila motor DC memiliki magnet tetap pada stator, motor stepper mempunyai magnet tetap pada rotor. Adapun spesifikasi dari motor stepper adalah banyaknya fasa, besarnya nilai derajat per step, besarnya *volt* tegangan catu untuk setiap lilitan, dan besarnya arus yang dibutuhkan untuk setiap lilitan. Motor stepper tidak dapat bergerak sendiri secara kontinyu, tetapi bergerak secara diskrit per-step sesuai dengan spesifikasinya. Untuk bergerak dari satu step ke step berikutnya diperlukan waktu dan menghasilkan torsi yang besar pada kecepatan rendah. Salah satu karakteristik motor stepper yang penting yaitu adanya torsi penahan, yang memungkinkan motor stepper menahan posisinya yang berguna untuk aplikasi motor stepper dalam yang memerlukan keadaan start dan stop. Setiap Motor Stepper memiliki karakteristiknya yang secara umum meliputi:

- Tegangan

Tiap motor stepper mempunyai tegangan rata-rata yang tertulis pada tiap unitnya atau tercantum pada datasheet masing-masing motor stepper. Tegangan rata-rata ini harus diperhatikan dengan seksama karena bila melebihi dari tegangan rata-rata ini akan menimbulkan panas yang menyebabkan kinerja putarannya tidak maksimal atau bahkan motor stepper akan rusak dengan sendirinya

- Resistansi

Resistansi per lilitan adalah karakteristik yang lain dari motor stepper. Resistansi ini akan menentukan arus yang

mengalir, selain itu juga akan mempengaruhi torsi dan kecepatan maksimum dan motor stepper.

- Derajat per step

Besarnya derajat putaran per step adalah parameter terpenting dalam pemilihan motor stepper karena akan menentukan ukuran langkah gerakan yang paling kecil (resolusi). Tiap-tiap motor stepper mempunyai spesifikasi masing-masing, antara lain 0.72° per step, 1.8° per step, 3.6° per step, 7.5° per step, 15° per step, dan bahkan ada yang 90° per step. Dalam pengoperasiannya kita dapat menggunakan 2 prinsip yaitu *full step* atau *half step*. Dengan *full step* berarti motor stepper berputar sesuai dengan spesifikasi derajat per stepnya, sedangkan *half step* berarti motor stepper berputar setengah derajat per step dari spesifikasi motor stepper tersebut.

Motor stepper sendiri juga memiliki jenisnya masing-masing, yaitu berdasarkan struktur rotor dan statornya meliputi: *Motor Stepper Variabel Reluctance*, *Motor Stepper Permanent Magnet*, dan *Motor Stepper Hybrid* yang kemudian digolongkan lagi menjadi unipolar dan bipolar. Pada Gambar 2.8 akan disajikan bentuk dari Motor Stepper.

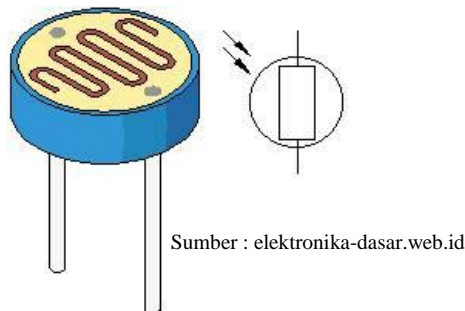


Sumber : partner3d.com

Gambar 2.8 Bentuk Motor Stepper.

2.6 Sensor LDR (*Light Dependent Resistor*)

Sensor LDR merupakan salah satu jenis sensor cahaya yang masuk dalam kategori resistor, resistor jenis ini dapat mengalami perubahan resistansinya apabila mengalami perubahan penerimaan cahaya. Besarnya nilai hambatan pada *Sensor Cahaya LDR (Light Dependent Resistor)* tergantung pada besar kecilnya cahaya yang diterima oleh LDR itu sendiri. LDR sering disebut dengan alat atau sensor yang berupa resistor yang peka terhadap cahaya. Biasanya LDR terbuat dari *cadmium sulfida* yaitu merupakan bahan semikonduktor yang resistansinya berubah-ubah menurut banyaknya cahaya (sinar) yang mengenainya. Resistansi LDR pada tempat yang gelap biasanya mencapai sekitar $10\text{ M}\Omega$, dan ditempat terang LDR mempunyai resistansi yang turun menjadi sekitar $150\ \Omega$. Seperti halnya resistor konvensional, pemasangan LDR dalam suatu rangkaian sama persis seperti pemasangan resistor biasa. Simbol LDR dapat dilihat seperti pada Gambar 2.9.



Gambar 2.9 Bentuk Fisik dan Simbol Sensor LDR

LDR termasuk dalam sebuah sensor atau komponen elektronik yang berarti ia memiliki karakteristiknya sendiri, yaitu *Laju Recovery* dan *Respon Spektral*, berikut penjelasannya :

- Laju *Recovery* Sensor Cahaya LDR (*Light Dependent Resistor*)

Bila sebuah “Sensor Cahaya LDR (*Light Dependent Resistor*)” dibawa dari suatu ruangan dengan level kekuatan cahaya tertentu ke dalam suatu ruangan yang gelap, maka bisa kita amati bahwa nilai resistansi dari LDR tidak akan segera berubah resistansinya pada keadaan ruangan gelap tersebut. Namun LDR tersebut hanya akan bisa mencapai harga di kegelapan setelah mengalami selang waktu tertentu. Laju *recovery* merupakan suatu ukuran praktis dan suatu kenaikan nilai resistansi dalam waktu tertentu. Harga ini ditulis dalam K/detik, untuk LDR tipe arus harganya lebih besar dari 200K/detik(selama 20 menit pertama mulai dari level cahaya 100 lux), kecepatan tersebut akan lebih tinggi pada arah sebaliknya, yaitu pindah dari tempat gelap ke tempat terang yang memerlukan waktu kurang dari 10 ms untuk mencapai resistansi yang sesuai dengan level cahaya 400 lux.

- Respon Spektral Sensor Cahaya LDR (*Light Dependent Resistor*)

Sensor Cahaya LDR (*Light Dependent Resistor*) tidak mempunyai sensitivitas yang sama untuk setiap panjang gelombang cahaya yang jatuh padanya (yaitu warna). Bahan yang biasa digunakan sebagai penghantar arus listrik yaitu tembaga, aluminium, baja, emas dan perak. Dari kelima bahan tersebut tembaga merupakan penghantar yang paling banyak, digunakan karena mempunyai daya hantar yang baik

2.7 Mbed OS

Mbed OS merupakan sebuah *Operating System* yang dirancang secara khusus untuk mikrokontroler berbasis ARM Cortex-M 32-bit. Dalam aplikasinya Mbed OS sudah banyak mendukung banyak mikrokontroller dari berbagai vendor. Mbed

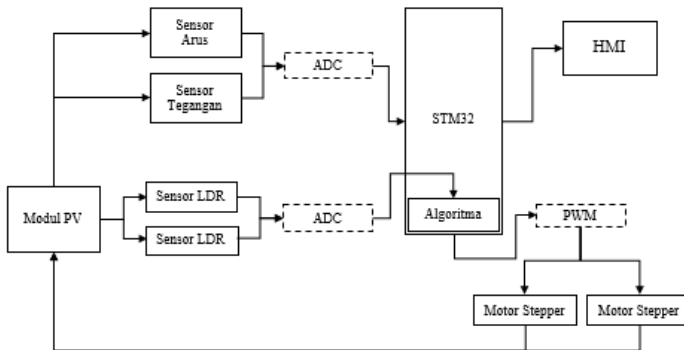
OS sendiri dibuat dengan menggunakan bahasa C dan C++ sehingga program yang dikembangkan dengan OS ini lebih mudah dan dapat di-*export* pada beberapa IDE khusus pengembangan mikrokontroler. Mbed OS juga dapat meng-*import* program dari IDE lain ke dalam sistemnya. Untuk melakukan tugas tersebut Mbed OS ini harus berjalan secara Online

BAB III

METODOLOGI PENELITIAN

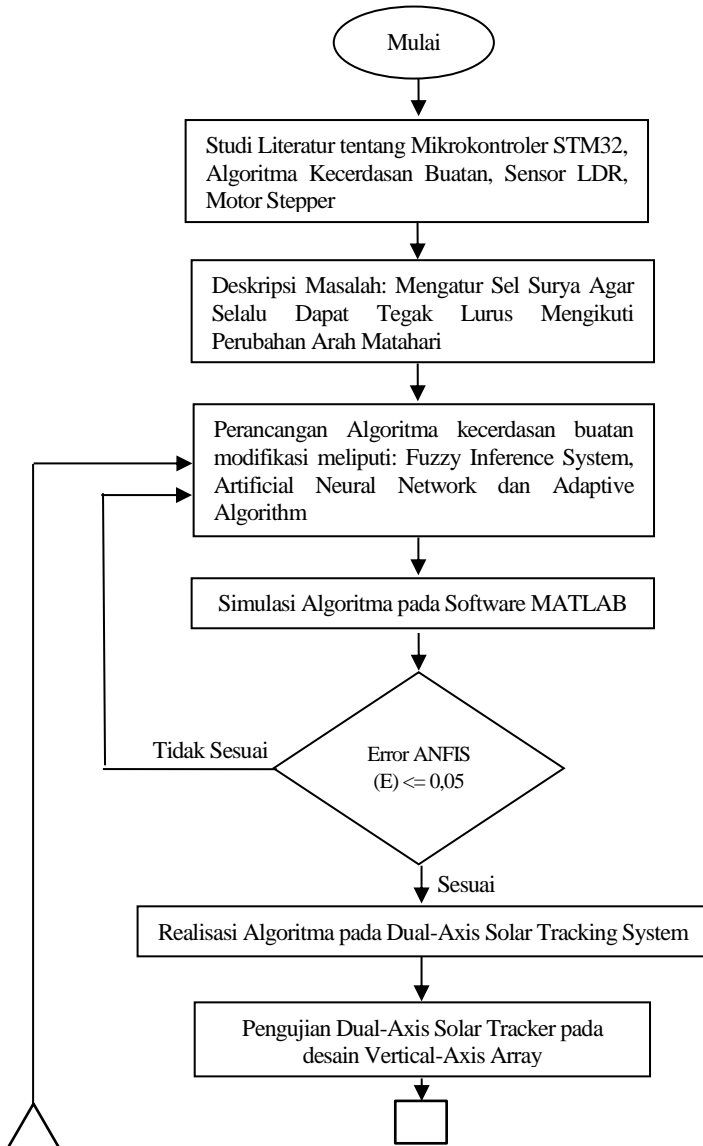
3.1 Flowchart Penelitian

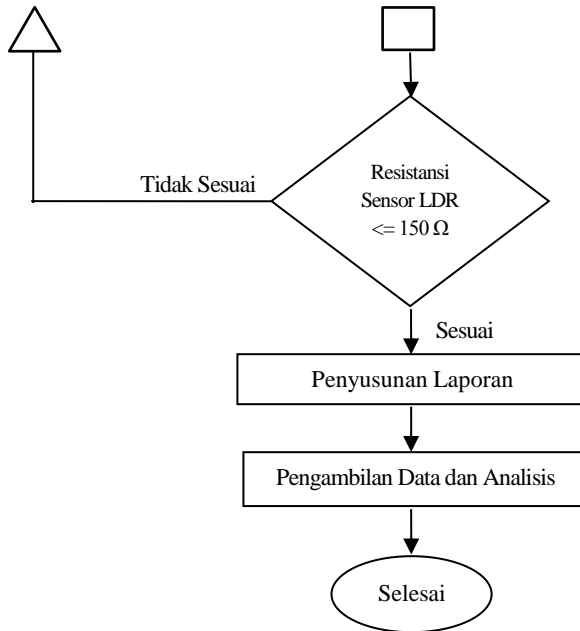
Pada perancangan tugas akhir ini memiliki keterkaitan dengan sistem yang lainnya. Adapun blok diagram sistem keseluruhan alat dijelaskan pada Gambar 3.1 berikut:



Gambar 3. 1 Diagram Blok Keseluruhan Sistem *Dual-Axis Solar Tracker*

Dari blok diagram keseluruhan sistem pada Gambar 3.1 terdapat bagian Algoritma yang terletak didalam mikrokontroller STM32 yang akan menjadi pokok bahasan kemudian, dimana bagian tersebut merupakan perancangan algoritma kecerdasan buatan modifikasi dari ANFIS (Adaptive Neuro Fuzzy Inference System) yang nantinya akan diimplementasikan pada sistem *dual-axis solar tracker*. Algoritma ini akan dilakukan modifikasi, dimana maksud modifikasi disini ialah menjadikan proses algoritma ANFIS ini menjadi semi-otomatis dalam pencarian bobot, bias dan *learning rate*. Adapun pengerjaan algoritma tersebut digambarkan dalam bentuk diagram alir atau *flowchart* seperti pada Gambar 3.2 berikut:





Gambar 3.2 *Flowchart* Penelitian

Dari *flowchart* pada Gambar 3.2 dapat dijelaskan secara sederhana dalam tahapan-tahapan sebagai berikut:

A. Studi Literatur

Penelitian ini dimulai dengan melakukan studi literatur. Literatur yang digunakan dapat berupa buku teks, paper, prosiding, dan laman internet dengan topik yang berhubungan dengan Mikrokontroler STM32, Algoritma Kecerdasan Buatan, Sensor LDR, Sensor Tegangan dan Sensor Arus, juga Motor Stepper

B. Deskripsi Masalah

Pengaturan Sel Surya agar selalu dapat tegak lurus mengikuti perubahan arah matahari sehingga efisiensi konversi daya diharapkan menjadi lebih optimal

C. Perancangan Algoritma Kecerdasan Buatan Modifikasi

Pada tahap ini Algoritma Kecerdasan Buatan mulai dirancang, Langkah yang dibutuhkan dalam merancang algoritma kecerdasan buatan ANFIS (*Adaptive Neuro Fuzzy Inference System*) ialah dengan merancang terlebih dahulu algoritma *Fuzzy Inference System*, lalu ke *Artificial Neural Network* dan kemudian digabungkan menjadi satu dengan *Adaptive Algorithm*.

D. Simulasi Algoritma pada Software MATLAB

Pada tahapan ini semua algoritma yang sudah dirancang sebelumnya disimulasikan dengan software Matlab dengan memanfaatkan ANFIS *Toolbox*-nya sebagai simulator untuk melihat output yang dihasilkan oleh algoritma ketika mendapati sebuah inputan tertentu. Hasil dari simulasi ini ialah untuk melihat nilai *error*(E) yang digunakan sebagai acuan untuk kondisi pemberhentian dari pembelajaran sistem ANFIS. Toleransi *error* yang digunakan ialah 0,05

E. Realisasi dan Algoritma pada *Dual-Axis Solar Tracking System*

Pada tahap ini algoritma yang sudah disimulasikan dan menunjukkan output yang diinginkan, selanjutnya akan direalisasikan pada mikrokontroller STM32 untuk menggerakkan PV sesuai dengan rotasi yang didapatnya.

F. Pengujian *Dual-Axis Solar Tracker* Pada Desain Vertical Axis Array

Pada Tahap ini algoritma akan diuji pada sistem PV untuk mengetahui nilai yang diinginkan seperti nilai Resistansi Sensor LDR yang nantinya harus kurang dari 150 Ω yang tandanya sel surya menghadap tegak lurus terhadap sinar matahari

G. Pengambilan Data dan Analisis

Pada Tahap ini Data sensor LDR dan data dari Sensor Arus dan Tegangan diambil untuk selanjutnya dianalisa sejauh mana performansi konversi sistem PV mengalami peningkatan

H. Penyusunan Laporan

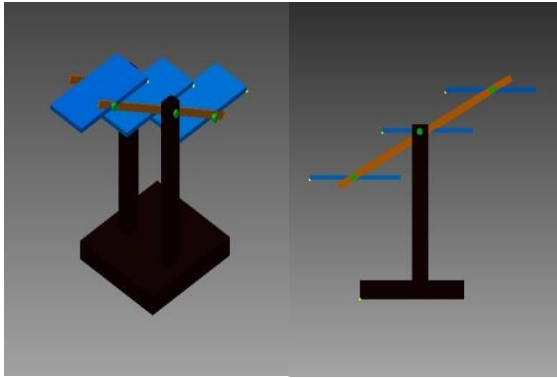
Pada tahapan ini, semua data baik hasil analisa atau hasil *monitoring* yang telah didapat dari tahap-tahap sebelumnya dikumpulkan menjadi satu untuk kemudian disusun menjadi Laporan Tugas Akhir

3.2 Perancangan Sistem

Pada perancangan sistem ini akan dijelaskan dalam poin-poin berikut ini:

3.2.1 Perancangan Algoritma ANFIS

Perancangan Algoritma ANFIS ini dikhususkan untuk *dual-axis solar tracker* dengan desain *vertical-axis array*, yang mana beberapa Sel PV diletakkan secara vertikal bertumpukan. Dikatakan *dual-axis* karena terdapat dua axis dalam desain ini. *Axis* pertama mengatur pergerakan setiap sel PV agar dapat mengikuti dan sejajar dengan arah datangnya matahari mulai dari timur hingga barat, sedangkan *axis* kedua ialah bagian yang mengatur pergerakan seluruh sel PV agar antar sel PV satu dengan yang lain tidak saling bersinggungan, sehingga semua sel PV dapat menyerap cahaya matahari secara optimal [15]. Lalu sebagai penggeraknya digunakan dua buah motor stepper untuk kedua *axis* yang saling bekerja sama menjalankan tugasnya [16]. Pada Gambar 3.3 akan disajikan gambaran desain untuk lebih jelasnya.



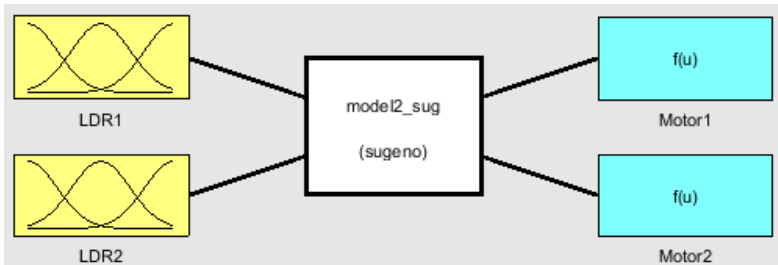
Gambar 3. 3 (a) Desain Tampak Depan(kiri) (b) Tampak Samping(kanan)

Desain tersebut juga memberi opsi untuk dapat menambah sel PV sesuai dengan kebutuhan daya output yang diinginkan. Dalam sistem kali ini, kebutuhan daya keluaran ialah berkisar 5,7 kWh [17], yang rencananya digunakan sebagai sumber energi untuk Smart Classroom pada Ruangan A300 di Departemen Teknik Instrumentasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember.

Dengan desain yang sudah dijelaskan, dimana sel PV bergerak dalam dua DOF(*Degree of Freedom*) yaitu Timur dan Barat, maka akan digunakan dua sensor LDR pada sistem ini sebagai penjajak pergerakan matahari. Dua sensor LDR ini akan diletakkan pada sel PV yang terletak dibagian tengah. Darisini perancangan algoritma dapat dimulai.

Perancangan algoritma diawali dengan menentukan jumlah input yang akan diberlakukan pada ANFIS. Diketahui terdapat dua input disini yaitu dua buah sensor LDR sebelumnya, sebagai LDR1(Timur) dan LDR2(Barat). Lalu sebagai outputnya ialah berupa *duty cycle* yang akan diterapkan pada dua buah motor sebagai Motor1(*Axis1*) dan Motor2(*Axis2*).

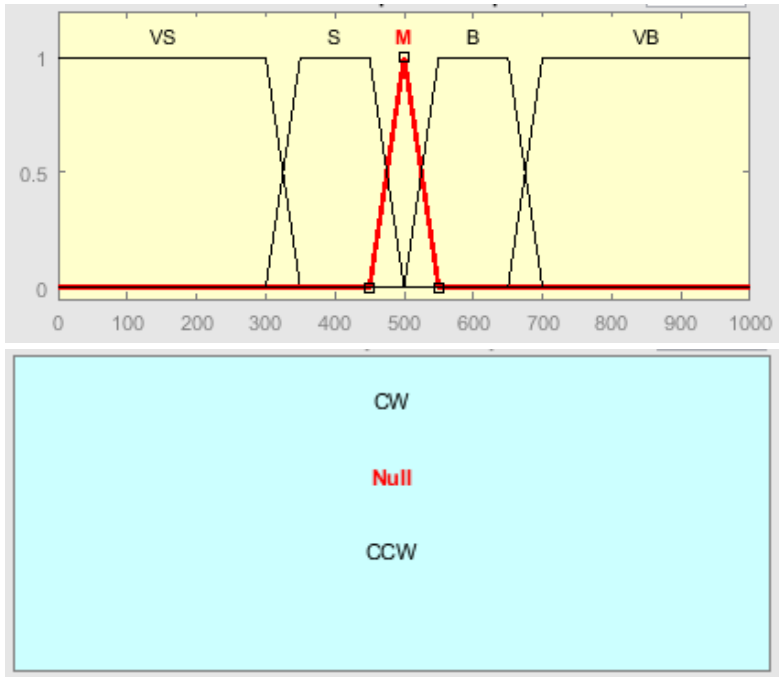
Selanjutnya dari Input dan Output tersebut dirancanglah Desain Algoritma Fuzzy dengan menggunakan bantuan *Fuzzy Logic Designer* pada software Matlab. Pada Gambar 3.4 akan dipaparkan desain dari algoritma fuzzy yang dibuat.



Gambar 3.4 Desain Algoritma Fuzzy

Desain pada Gambar 3.4 dibuat dengan menggunakan metode TSK (Takagi-Sugeno-Kang) dimana hasil dari metode ini ialah berupa konstanta jika orde nol atau persamaan linear jika orde satu. Hal ini akan memudahkan dalam perancangan Algoritma ANFIS dikarenakan dalam metode tersebut inferensinya tanpa perlu menggunakan implikasi maupun agregasi. Setelah desain algoritma fuzzy sudah dirancang, maka selanjutnya ialah menentukan Variabel dan Himpunan Fuzzy dari input maupun output algoritma fuzzy tersebut. Untuk Input, Variabel Fuzzy berupa LDR1 dan LDR2, sedangkan untuk Output, Variabel Fuzzy berupa Motor1 dan Motor2. Selanjutnya untuk tiap Input, terdapat lima Himpunan Fuzzy yang digunakan, yaitu *Very Small*(VS), *Small*(S), *Medium*(M), *Big*(B), dan *Very Big*(VB). Sedangkan untuk Output, terdapat tiga Himpunan Fuzzy yaitu *ClockWise*(CW), *Null*, dan *Counter ClockWise*(CCW). Setelah variabel maupun himpunannya ditentukan, maka selanjutnya ialah menentukan Derajat Keanggotaan dari tiap Himpunan. Hal

tersebut dapat dirancang dengan menggunakan bantuan *Membership Function Editor* pada Software Matlab. Pada Gambar 3.5 akan dijelaskan mengenai desain dari Derajat Keanggotaan, Variabel, dan himpunan dari algoritma Fuzzy untuk sistem ini.



Gambar 3.5 (a) Derajat Keanggotaan Variabel Input(atas) (b) Derajat Keanggotaan Variabel Output(bawah)

Dari Gambar 3.5, data dapat disajikan dalam bentuk tabel yang dapat menjelaskan dari Variabel dan Himpunan Fuzzy hingga Fungsi Keanggotaan dengan Range dan Parameternya. Sehingga lebih mudah dalam hal pembacaan data tersebut. Pada Tabel 3.1 akan dijelaskan mengenai tabel yang menyajikan data-data tersebut.

Tabel 3.1 Variabel, Himpunan, Range, Parameter, Fungsi Keanggotaan Fuzzy

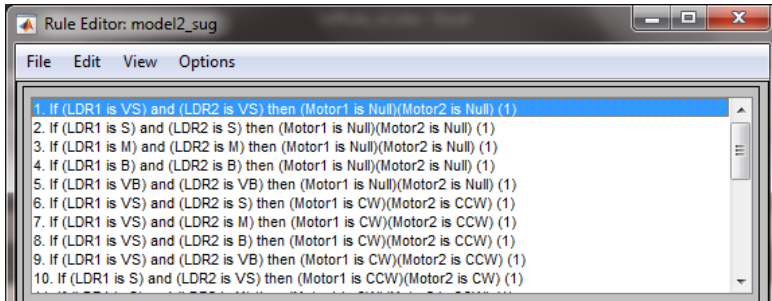
Fungsi	Variabel	Nama Himpunan Fuzzy	Range	Parameter	Fungsi Keanggotaan
Input	LDR1	VS	[0-1000]	[0-350]	Linear Turun
		S		[300-500]	Trapesium
		M		[450-550]	Segitiga
		B		[500-700]	Trapesium
		VB		[650-1000]	Linear naik
	LDR2	VS	[0-1000]	[0-350]	Linear Turun
		S		[300-500]	Trapesium
		M		[450-550]	Segitiga
		B		[500-700]	Trapesium
		VB		[650-1000]	Linear naik
Output	Motor1	CW	[-180-180]	[0-180]	Linear
		Null		[-10-10]	
		CCW		[-180-0]	
	Motor2	CW	[-180-180]	[0-180]	
		Null		[-10-10]	
		CCW		[-180-0]	

Setelah Derajat Keanggotaan sudah ditentukan. maka prose Fuzifikasi selesai. Selenjutnya masuk pada proses *Inference Rule*, yaitu proses mendapatkan output dari *Rules* yang sudah ada. Dalam proses ini tujuan yang dicapai ialah agar masing-masing sel PV dapat bekerja secara optimal, hal tersebut dapat diartikan ketika sel PV yang terdapat sensor LDR bernilai kurang dari sama dengan 150 Ohm. Maka dari tabel derajat kebebasan diatas didapat tabel *inference rule* seperti pada Tabel 3.2 berikut.

Tabel 3.2 Inference Rule Fuzzy

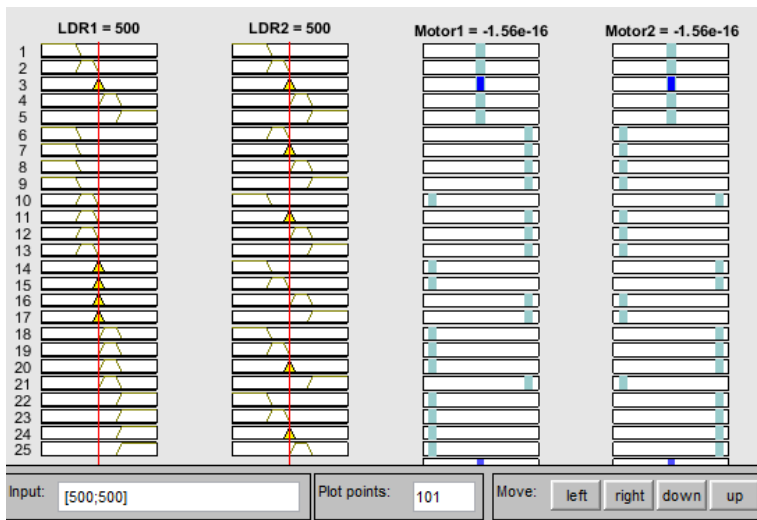
		LDR1				
		VS	S	M	B	VB
LDR2	VS	Null	CCW, CW	CCW, CW	CCW, CW	CCW, CW
	S	CW, CCW	Null	CCW, CW	CCW, CW	CCW, CW
	M	CW, CCW	CW, CCW	Null	CCW, CW	CCW, CW
	B	CW, CCW	CW, CCW	CW, CCW	Null	CCW, CW
	VB	CW, CCW	CW, CCW	CW, CCW	CW, CCW	Null

Setelah *Inference Rule* didata seperti diatas, maka dengan bantuan *Rule Editor* pada software Matlab, model fuzzy yang sudah dibuat sebelumnya dapat ditemukan nilai nya dengan cara meng-*plot rules* nya, karena pada algoritma Fuzzy ini dibuat dengan metode TSK maka logic atau kondisi yang digunakan ialah logic AND. Pada Gambar 3.6 akan dipaparkan pem-plottingan dari Rules-rules sebelumnya.

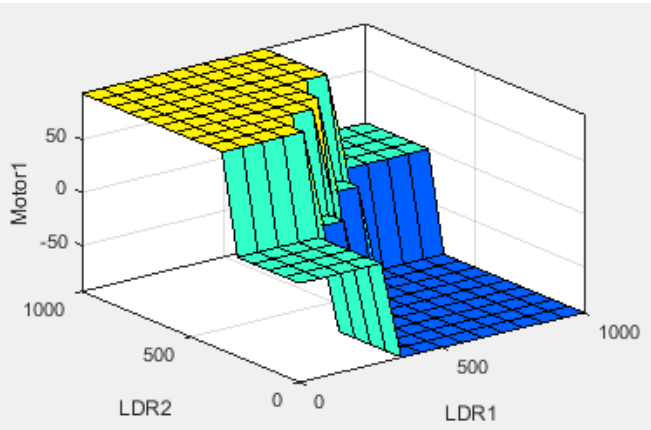


Gambar 3.6 Plotting Inference Rule pada Rule Editor

Setelah *Inference Rule* yang sudah ditentukan sebelumnya sudah terplot semua, maka output dapat dilihat dengan menggunakan bantuan *Rule Viewer* dan *Surface Viewer* pada Software Matlab seperti pada Gambar 3.7 dan Gambar 3.8



Gambar 3.7 Nilai Input dan Output pada Rule Viewer



Gambar 3.8 Input dan Output pada *Surface Viewer*

Dari kedua tool sebelumnya, maka output dapat disajikan dalam bentuk tabel yang berisikan semua kemungkinan input dan output nya. Pada Tabel 3.3 akan disajikan sampel data output dari kemungkinan beberapa input yang diperoleh dengan menggunakan bantuan software MATLAB.

Tabel 3.3 Sampel data Output Fuzzy(Z)

LDR1	LDR2	Motor1	Motor2
0-300	0-300	0,00	0,00
0-300	301	1,87	-1,87
0-300	302	3,74	-3,74
0-300	303	5,61	-5,61
0-300	304	7,47	-7,47
0-300	305	9,34	-9,34
0-300	306	11,21	-11,21
0-300	307	13,07	-13,07
0-300	308	14,94	-14,94
0-300	309	16,81	-16,81

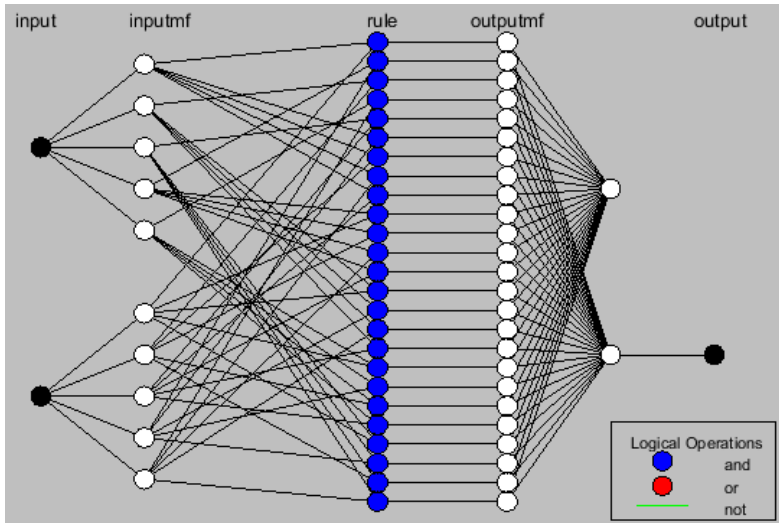
0-300	310	18,68	-18,68
0-300	311	20,54	-20,54
0-300	312	22,41	-22,41
0-300	313	24,28	-24,28
0-300	314	26,14	-26,14
0-300	315	28,01	-28,01
0-300	316	29,88	-29,88
0-300	317	31,74	-31,74
0-300	318	33,61	-33,61
0-300	319	35,48	-35,48
0-300	320	37,35	-37,35
0-300	321	39,21	-39,21
0-300	322	41,08	-41,08
0-300	323	42,95	-42,95
0-300	324	44,81	-44,81
0-300	325	46,68	-46,68
0-300	326	48,55	-48,55
0-300	327	50,41	-50,41
0-300	328	52,28	-52,28
0-300	329	54,15	-54,15
0-300	330	56,02	-56,02
0-300	331	57,88	-57,88
0-300	332	59,75	-59,75
0-300	333	61,62	-61,62
0-300	334	63,48	-63,48
0-300	335	65,35	-65,35
0-300	336	67,22	-67,22
0-300	337	69,08	-69,08
0-300	338	70,95	-70,95

0-300	339	72,82	-72,82
0-300	340	74,69	-74,69
0-300	341	76,55	-76,55
0-300	342	78,42	-78,42
0-300	343	80,29	-80,29
0-300	344	82,15	-82,15
0-300	345	84,02	-84,02
0-300	346	85,89	-85,89
0-300	347	87,75	-87,75
0-300	348	89,62	-89,62
0-300	349	91,49	-91,49
0-300	350	93,36	-93,36
0-300	351-1000	93,36	-93,36

Untuk lebih lengkap mengenai output dari fuzzy dapat dilihat pada Lampiran A. Dapat diketahui dari tabel tersebut bahwa output minimal dan maksimal fuzzy ialah -93,4 dan 93,4. Nilai ini berbeda dengan harapan perancangan yaitu -180 dan 180. Faktor penurunan nilai ini disebabkan oleh pengkonversian logika fuzzy dari metode mamdani ke metode sugeno atau TSK. Selanjutnya output atau Z fuzzy itulah yang akan diproses oleh Algoritma selanjutnya yaitu JST (Jaringan Saraf Tiruan) untuk dicari nilai ketetapanannya. Maka proses perancangan algoritma fuzzy telah selesai. Selanjutnya ialah merancang algoritma JST atau ANN(*Adaptive Neural Network*) berdasarkan algoritma fuzzy yang sudah dirancang sebelumnya.

Perancangan Algoritma ANNawali dengan membuat desain struktur *Neural Network*-nya. Disini pembuatan desain tersebut dapat menggunakan bantuan *tool Neuro-Fuzzy Designer*

pada Software Matlab. Hasil dari desain tersebut dapat dilihat pada Gambar 3.9



Gambar 3.9 Desain Struktur ANFIS

Setelah struktur ANFIS sudah didesain, maka selanjutnya ialah melakukan *training* and *testing* pada desain pada Gambar 3.9. Dengan menggunakan software Matlab maka *training* dan *testing* dapat dilakukan secara otomatis, namun dikarenakan Software Matlab hanya mampu melakukan *Training and Testing* pada desain *single output* dan *double input* sedangkan pada desain ini ialah *double input double output*, maka *training and testing* untuk mencari nilai Bias dan Beban dilakukan secara Semi-Otomatis dengan bantuan Software Excel. Hal tersebutlah yang menjadikan algoritma ini diberikan label modifikasi pada penamaannya. *Training* dan *testing* disini akan dilakukan dengan menggunakan metode *backpropagation*. Hal ini dikarenakan dengan metode

tersebut dapat dicapainya *adaptive algorithm*. Adapun algoritma dari *backpropagation* sendiri ialah sebagai berikut (Fauset 1994):

Langkah 0 : Inisialisasi bobot-bobot, konstanta laju pelatihan (α), toleransi *error* atau nilai bobot (bila menggunakan nilai bobot sebagai kondisi berhenti) atau set maksimal epoch (jika menggunakan banyaknya epoch sebagai kondisi berhenti).

Langkah 1 : Selama kondisi berhenti belum dicapai, maka lakukan langkah ke-2 hingga langkah ke-9. Disini kondisi pemberhentian ialah ketika E kurang dari atau sama dengan 0,05

Langkah 2 : Untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-8.

TAHAP 1 (*FeedForward*)

Langkah 3 : Setiap unit input x_i (dari unit ke-1 hingga unit ke-n pada lapisan input) mengirimkan sinyal input ke setiap input yang berada pada lapisan tersembunyi.

Langkah 4 : Masing-masing unit di lapisan tersembunyi (dari unit ke-1 hingga unit ke-p) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan dengan biasnya :

$$Z_{in_j} = V_{oj} + \sum_{i=1}^n X_i V_{ij} \quad (3.1)$$

Kemudia menghitung sesuai nilai aktivasi yang diguakan:

$$Z_j = f(Z_{in_j}) \quad (3.2)$$

Karena yang digunakana disini ialah fungsi aktivasi sigmoid biner maka persamaan (8) menjadi:

$$Z_j = \frac{1}{1 + \exp(-z_{in_j})} \quad (3.3)$$

Langkah 5 : Masing-masing unit output (y_k , $k=1,2,3,\dots,m$) dikalikan dengan bobot dan dijumlahkan serta ditambahkan dengan biasnya.

$$Y_{in_k} = W_{ok} + \sum_{j=1}^p Z_j W_{jk} \quad (3.4)$$

Kemudian menghitung sesuai nilai aktivasi yang digunakan:

$$Y_k = f(Y_{in_k}) \quad (3.5)$$

TAHAP 2 (BackPropagation dan Error)

Langkah 6 : Masing-masing unit output (Y_k , $k=1,2,3,\dots,m$) menerima pola target t_k sesuai dengan pola masukan/input saat pelatihan dan kemudian informasi kesalahan/error lapisan output (δ_k) dihitung. δ_k dikirim ke lapisan dibawahnya dan digunakan untuk menghitung besarnya koreksi bobot dan bias (ΔW_{jk} dan ΔW_{ok}) antara lapisan tersembunyi dengan lapisan output :

$$\delta_k = (t_k - y_k) f'(Y_{in_k}) \quad (3.6)$$

Karena $f'(Y_{in_k}) = Y_k$, menggunakan fungsi sigmoid bipolar maka:

$$\begin{aligned} f'(Y_{in_k}) &= f(Y_{in_k}) (1 - f(Y_{in_k})) \\ &= Y_k (1 - Y_k) \end{aligned} \quad (3.7)$$

Hitung suku perubahan bobot W_{jk} (yang akan digunakan untuk merubah bobot W_{jk}) dengan laju pelatihan α :

$$\Delta W_{kj} = \alpha \delta_k Z_j \quad (3.8)$$

Hitung perubahan bias:

$$\Delta W_{ok} = \alpha \delta_k W_{jk} \quad (3.9)$$

Langkah 7 : Pada setiap unit di lapisan tersembunyi (dari unit ke-1 hingga ke-p; $i=1 \dots n$; $k=1 \dots m$) dilakukan perhitungan informasi kesalahan lapisan tersembunyi (δ_j). δ_j kemudian digunakan untuk menghitung besar koreksi bobot dan bias (ΔV_{ji} dan ΔV_{jo}) antara lapisan input dan lapisan tersembunyi.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \quad (3.10)$$

Selanjutnya dikalikan dengan turunan dari fungsi aktivasinya untuk menghitung galat.

$$\delta_j = \delta_{in_j} f'(Y_{in_k}) \quad (3.11)$$

Hitung suku perubahan bobot V_{ji} (yang digunakan untuk perbaikan bobot V_{ji}).

$$\Delta V_{ij} = \propto \delta_j X_i \quad (3.12)$$

Hitung perubahan bias (untuk memperbaiki V_{jo}).

$$\Delta V_{oj} = \propto \delta_j \quad (3.13)$$

TAHAP 3 (Update Bobot dan Bias)

Langkah 8 : Masing-masing unit output/keluaran (y_k , $k=1,2,3,\dots,m$) dilakukan pengupdetan bias dan bobotnya ($j = 0,1,2,\dots,p$) sehingga menghasilkan bobot dan bias baru :

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (3.14)$$

Demikian juga untuk setiap unit tersembunyi mulai dari unit ke-1 sampai dengan unit ke-p dilakukan pengupdatean bobot dan bias:

$$V_{jk}(\text{baru}) = V_{jk}(\text{lama}) + \Delta V_{jk} \quad (3.15)$$

Langkah 9 : Uji kondisi berhenti (akhir iterasi)

Daftar Notasi:

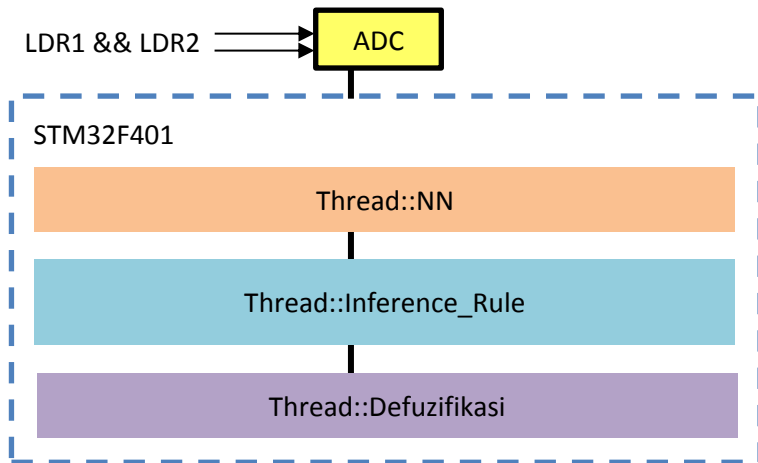
X_p	= Pola masukan pelatihan ke-p, $p=1,2,\dots,p\leq 1$
X_p	= (X ₁ , X ₂ , X ₃ ,...,X _n)
tp	= pola keluaran target dari pelatihan
tp	= (t ₁ , t ₂ , t ₃ , . . . ,t _n)
x_i	= Unit ke-I pada lapisan masukan
X_i	= nilai aktivasi dari unit X _i
Z_j	= Unit ke-j pada lapisan tersembunyi
Z_{in}_j	= keluaran untuk unit Z _j
z_j	= nilai aktivasi dari unit Z _j
Y_k	= unit ke -k pada lapisan keluaran
Y_{in}_k	= net masukan untuk unit Y _k
Y_k	= nilai aktivasi dari unit Y _k
W_{ko}	= nilai penimbang sambungan pada bias untuk unit Y _k
W_{kj}	= nilai penimbang sambungan dari Z _j ke unit Y _k
ΔW_{kj}	= selisih antara W _{kj} (t) dengan W _{kj} (t+1)
V_{io}	= nilai penimbang sambungan pada bias untuk unit Z _i
V_{ij}	= nilai penimbang sambungan dari unit X _i ke unit Z _i
ΔW_{ij}	= selisih antara V _{ij} (t) dengan V _{ij} (t+1)
δ_k	= faktor pengaturan nilai penimbang sambungan pada lapisan keluaran
δ_j	= faktor pengaturan nilai penimbang sambungan pada lapisan tersembunyi
α	= konstanta laju pelatihan (learning rate) $0 < \alpha < 1$
E	= Total galat

Dengan diperolehnya tiap bobot dan bias berdasarkan algoritma diatas, dimana E harus kurang atau sama dengan 0,05 untuk batas atas, dan lebih besar atau sama dengan -0,05 untuk batas bawah, yang dalam kasus ini ialah kondisi pemberhentian algoritma ANFIS ini, maka selanjutnya ialah merealisasikan

algoritma tersebut kedalam bentuk listing code pada mikrokontroler yang digunakan yaitu pada nukleo STM32F401RE.

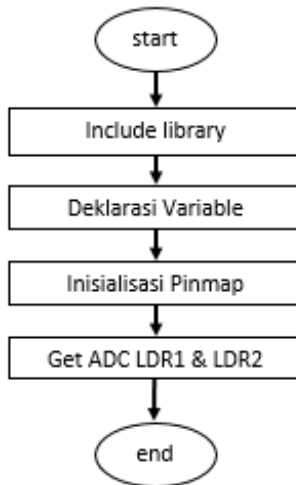
3.2.2 Realisasi Algoritma ANFIS

Langkah awal dalam realisasi ini ialah dengan merancang diagram blok fungsinya. Pada Gambar 3.10 ini akan dipaparkan diagram blok dalam merealisasikan algoritma ANFIS yang sudah dirancang pada *board* STM32F401.



Gambar 3.10 Diagram Blok Fungsi Algoritma ANFIS

Dalam diagram blok pada Gambar 3.10 diketahui MCU yang digunakan ialah STM32 keluarga F401. Digunakannya MCU ini dikarenakan penggunaan algoritma ANFIS yang membutuhkan pemrosesan cepat dalam pengolahan algoritmanya. Telah dibahas sebelumnya jika input dari algoritma ini berupa nilai ADC dari dua sensor LDR. Adapun *flowchart* untuk mendapatkan nilai ADC tersebut akan dijelaskan pada Gambar 3.11.



Gambar 3.11 Flowchart LDR pada ADC

Dari proses pada Gambar 3.11, dapat disajikan penjelasan dalam bentuk *pseudocode*, berikut *pseudocode* untuk mencari nilai ADC sensor LDR.

Pseudocode: get ADC value

Include mbed.h :Library

Initialitation of Mbed Pherpherals:

PC_2: Analog_in sebagai analog_LDR1

PC_3: Analog_in sebagai analog_LDR2

Declaration of Variable:

LDR1, LDR2: float

a: integer

LDR1 <- baca ADC analog_LDR1

LDR2 <- baca ADC analog_LDR2

Berdasarkan pada *pseudocode: get ADC value* diatas, maka akan didapat sebuah *output* nilai berupa ADC dari LDR dan

nilainya diletakkan pada variabel LDR1 dan LDR2. Setelah nilai ADC didapatkan maka selanjutnya ialah merancang algoritma ANFIS sebelumnya pada RTOS(*Real-Time Operating System*) di MCU yang digunakan. RTOS disini berfungsi sebagai sarana agar *task* atau tugas yang dirancang dapat berjalan secara *real-time*, atau lebih mudahnya agar *task-task* yang dirancang dapat berjalan bersamaan sehingga dapat mengefisiensi waktu yang digunakan dalam pemrosesan data. Dalam RTOS dikenal *Thread*. *Thread* sendiri merupakan sebuah fungsi yang dirancang khusus pada RTOS untuk menjalankan tugas tertentu secara *real-time*. Dalam hal ini terdapat tiga *thread* yang akan mewakili algoritma ANFIS, yaitu: *Neural Network* (NN), *Inference Rule* dan *Defuzifikasi*. Untuk thread pertama yaitu thread:NN akan disajikan dalam bentuk *flowchart* seperti pada Gambar 3.12. Sedangkan untuk penjelasan dalam bentuk *pseudocode* dapat dilihat pada *pseudocode: thread_NN*

Pseudocode: thread_NN

Include mbed.h, rtos.h :Library

Declaration of Variable:

Input1, Input2: float

Input1 <- LDR1

Input2 <- LDR2

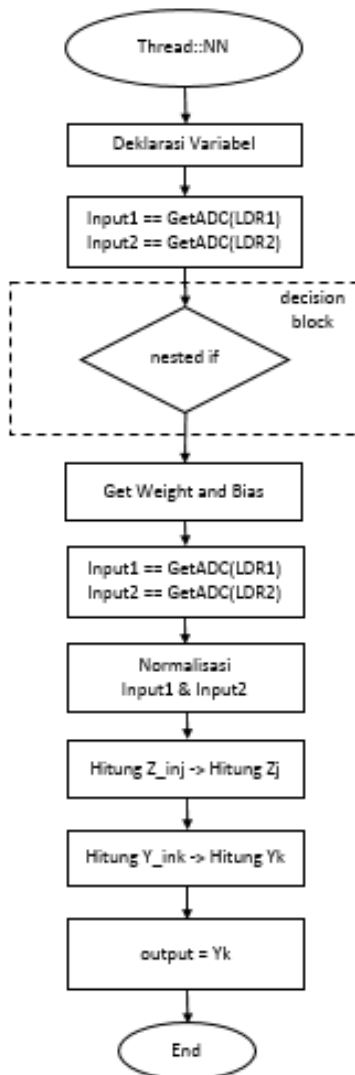
Block: Nested If

IF Input1 and Input2 = Value of Input1 or Input2 Then

Get Value: Weight, Bias

Normalisasi Input1 dan Input2

Hitung Yk



Gambar 3.12 Flowchart thread NN

Berdasarkan *pseudocode: thread_NN*, maka akan didapatkan sebuah output dari proses NN *feedforward*, yang mana data nilai bobot dan bias didapatkan berdasarkan tabel pada Lampiran A.

Selanjutnya setelah proses dari *thread::NN* selesai maka masuk pada *thread* berikutnya yaitu *thread::Inf_Rule*. Pada Gambar 3.13 akan disajikan *flowchart* dari *thread::InfRule*. Sedangkan untuk *pseudocode* nya yaitu *thread_InfRule* akan disajikan dibawah ini.

Pseudocode: thread_InfRule

Include mbed.h, rtos.h :Library

Declaration of Variable:

Input1, Input2, Motor1, Motor2: float

Checker1, checker2: integer

Input1 <- LDR1

Input2 <- LDR2

IF Input1 and Input2 = *Value of* Input1 or Input2 Then

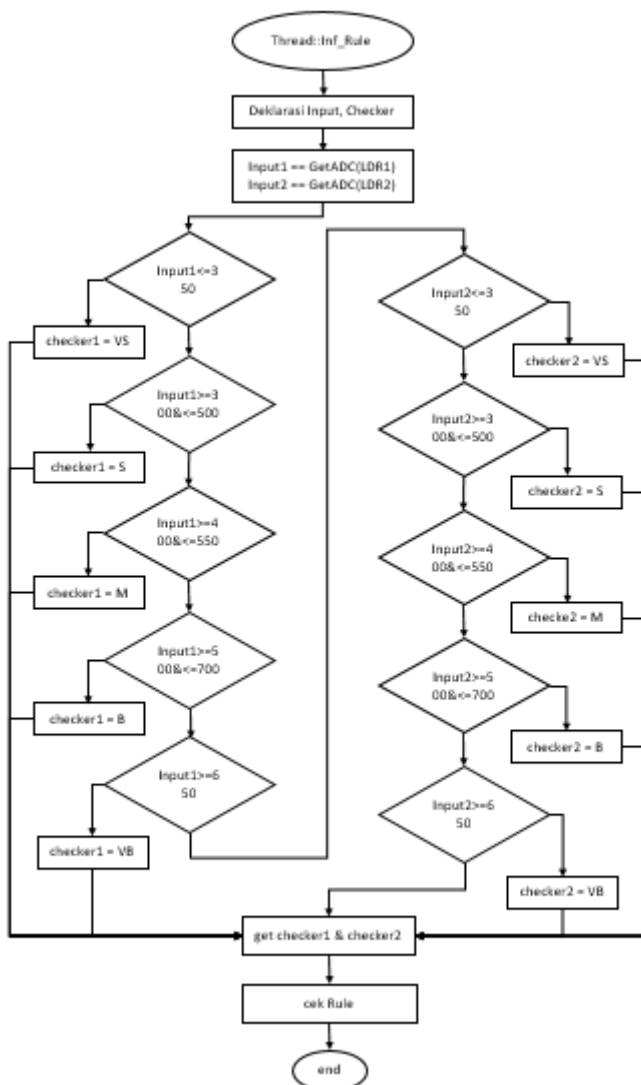
Checker1 or Checker2 = 1:VS;2:S;3:M;4:B;5:VB

Get Checker1 and Checker2

Check Rule()

Get temp_FinalValue:

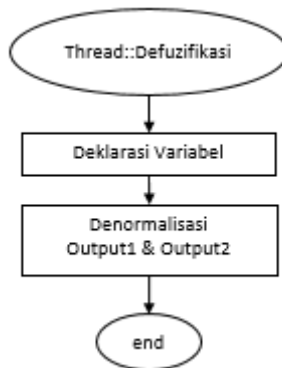
Motor1, Motor2



Gambar 3.13 Flowchart thread Inferensi Rule

Berdasarkan *pseudocode thread_InfRule*, maka akan didapatkan dua buah *output* berubah nilai akhir sementara yang nantinya dikonversi kedalam *duty cycle*, *output* tersebut ialah pada Variabel Motor1 dan Motor2.

Terdapat label sementara pada paragraf sebelumnya karena nilai tersebut masih perlu diproses lagi agar menjadi nilai akhir yang sebenarnya, yaitu dengan cara menormalisasinya ulang menjadi *range* yang diinginkan yaitu antara -93,4 sampai 93,4. Pada Gambar 3.14 akan disajikan *flowchart* *thread::Defuzifikasi*, sedangkan untuk *pseudocode*-nya akan disajikan setelah Gambar 3.14.



Gambar 3.14 Flowchart thread Defuzifikasi

Pseudocode: thread_Defuzifikasi

Include mbed.h, rtos.h :Library

Declaration of Variable:

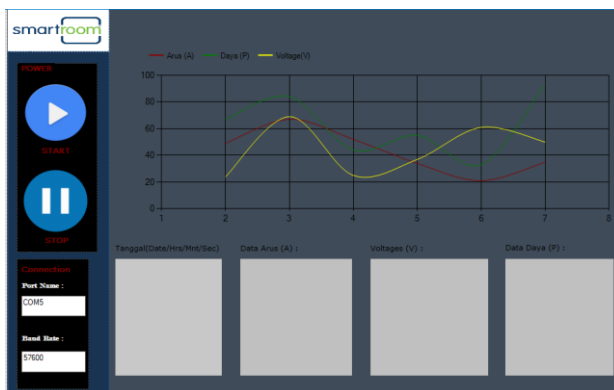
Output1, Output2: float

Normalisasi Motor1 dan Motor2

Output1 <- Motor1

Output2 <- Motor2

Setelah proses pada thread::Defuzifikasi selesai, maka akan dihasilkan dua buah output berupa nilai *duty cycle* yang diperlukan motor1 agar dapat menggerakkan PV tegak lurus terhadap arah matahari dan motor2 untuk dapat menggerakkan rangka agar antar PV dapat bekerja secara optimal. Sampai sini maka algoritma ANFIS selesai. Dengan didapatkannya titik kerja PV tersebut, maka selanjutnya ialah me-*sensing* arus dan tegangan untuk mendapatkan keluaran daya yang didapat. hal ini dilakukan dengan tujuan untuk mengetahui performa dari kinerja algoritma yang telah dibuat. Nilai nilai hasil disini dimonitoring dengan merancang HMI yang dikhususkan untuk penelitian ini, dimana nilai yang ditransfer ke HMI ialah bersifat nirkabel atau wireless dengan menggunakan hardware 3DR Radio Telematic sehingga monitoring dapat dilakukan dengan mudah dimanapun juga [15], selain itu juga hasil tersebut disimpan dalam memori card sebagai alternatif monitoring ketika terjadinya gangguan pada komunikasi data pada 3DR. Lalu untuk perancangan HMI dilakukan dengan menggunakan bantuan software Visual Studio C#. Berikut ialah gambar dari desain HMI yang dibuat

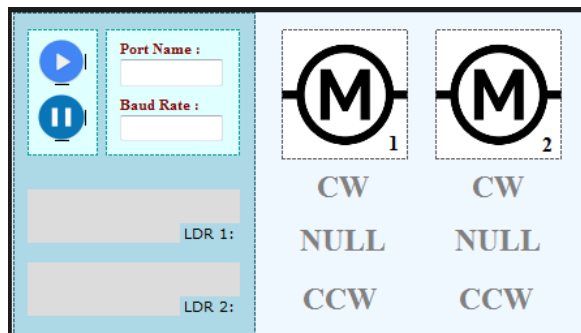


Gambar 3.15 Desain HMI khusus Smart Classroom

3.3 Eksperimen Algoritma ANFIS

Setelah algoritma selesai dirancang, maka untuk mengetahui algoritma tersebut sudah sesuai yang diinginkan atau belum sesuai, maka dilakukan eksperimen terhadap algoritma yang telah dirancang. Eksperimen tersebut ialah dengan cara membuat program berbasis OOP (*Object Oriented Programming*) sederhana, yang mana program tersebut akan menerima data ADC dari sensor LDR lewat komunikasi serial ke PC berbasis *wireless*. Program tersebut akan mempresentasikan nilai output dari algoritma ANFIS dan juga arah dari pergerakan motor1 dan motor2.

Pemberian input eksperimen dilakukan dengan cara memberi input pada sensor LDR berupa rangsangan cahaya, dalam kasus ini cahaya yang digunakan berasal dari cahaya *LED Flashlight* dari *smartphone*. *Flashlight* akan diarahkan ke sensor LDR1 untuk mensimulasikan matahari terbit dan diarahkan menuju sensor LDR2 untuk mensimulasikan matahari terbenam. Algoritma akan dikatakan berhasil jika output pada motor1 dan motor2 bernilai Null atau bisa dikatakan kedua motor tidak bergerak atau berputar. Pada gambar 3.16 akan disajikan desain program OOP yang dibuat untuk eksperimen ini.



Gambar 3.16 Desain program OOP khusus untuk eksperimen algoritma

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan Algoritma ANFIS

Adapun hasil perancangan algoritma ANFIS ialah dalam bentuk bobot unit dan bobot bias dari keseluruhan unit meliputi unit tersembunyi dan unit keluaran. Berikut ini merupakan sampel tabel dari bobot dan bias untuk beberapa tiap unit masukan.

Tabel 4.1 Daftar Bobot sambungan tiap unit dan Bobot bias

Input1	Input2	E	epoch	vij1	vij2	voj	wjk	wok	Yk	Tk
300	300	-0,005	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,741773	0
300	301	-0,0025	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,740565	1,87
300	302	8,6E-07	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,739358	3,74
300	303	0,0025	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,738152	5,61
300	304	0,00499	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,736947	7,47
300	305	0,005	217	0,950713	0,951211	0,930762	-0,14681	0,239969	5,590187	9,34
300	306	0,00499	216	0,950462	0,950955	0,929982	-0,11101	0,250695	7,450846	11,207
300	307	0,005	213	0,950282	0,950771	0,929426	-0,07573	0,261736	9,300455	13,074
300	308	0,00499	212	0,950171	0,950657	0,929082	-0,04064	0,273197	11,15488	14,941
300	309	0,00499	210	0,950127	0,950612	0,928945	-0,00593	0,285028	13,00345	16,808
300	310	0,00499	208	0,950148	0,950633	0,929009	0,028426	0,297239	14,84602	18,675
300	311	0,005	206	0,950232	0,95072	0,929269	0,062438	0,30984	16,68246	20,542
300	312	0,005	205	0,950377	0,950871	0,929719	0,096261	0,322899	18,51998	22,409
300	313	0,005	204	0,950583	0,951086	0,930357	0,12987	0,336416	20,35612	24,276
300	314	0,00499	203	0,950847	0,951362	0,931177	0,163252	0,350397	22,18914	26,143
300	315	0,005	201	0,951167	0,951699	0,932172	0,19627	0,364791	24,01033	28,01
300	316	0,00499	201	0,951545	0,952097	0,933346	0,22924	0,379749	25,83604	29,877

Dari tabel tersebut dapat diketahui banyak data sekaligus. Miasnya untuk input1 sebesar 300 dan input2 sebesar 300 maka didapatkan bobot sambungan dari tiap unit masukan menuju unit tersembunyi(vij) sebesar 0,95 untuk LDR1 dan 0,95 untuk LDR2, sedangkan bobot biasnya(voj) ialah 0,93. Lalu untuk bobot dari unit tersembunyi menuju keluaran(wjk) bernilai -0,18 begitupun juga bobot bias yang digunakan(wok) yaitu bernilai 0,22. Output yang ditargetkan berdasarkan kedua input tersebut ialah 0 sedangkan hasil dari Yk dengan menggunakan niali bobot-bobot tersebut ialah 3,74, sehingga dengan begitu nilai error yang

diperoleh ialah -0.005 untuk pola tersebut. Semua proses hingga nilai bobot-bobot tersebut ditemukan dilakukan dalam 1 kali pembelajaran Algoritma, atau bisa disebut 1 *epoch* hanya untuk pola tersebut. Dapat diketahui jikalau selisih nilai dari T_k dan Y_k sangatlah besar, perubahan nilai yang didapat ini disebabkan oleh beberapa faktor misalnya nilai aktivasi yang digunakan pada algoritma JST. Dimana nilai aktivasinya ialah berkisar antara 0 hingga 1 dengan menggunakan fungsi aktivasi sigmoid biner, dengan kecilnya range nilai aktivasi maka hasil dari unit tersembunyi pun akan mengecil hingga mempengaruhi nilai Y_k . Faktor selanjutnya ialah *learning rate* yang masih menggunakan angka random tanpa menggunakan *deep learning* yang mempengaruhi fluktuasi peng-*update*-an bobot dan bias. Lalu faktor terakhir ialah masihnya menggunakan cara semi-otomatis pada proses algoritmanya.

4.2 Penjelasan *Listing Code*

Seperti yang sudah dijelaskan pada bab sebelumnya, yang mana Algoritma ANFIS dalam bentuk *listing code* terbagi menjadi beberapa blok program, antara lain program ADC sensor LDR, lalu program algoritma ANFIS yang terbagi menjadi tiga *thread*. Adapun *listing code* untuk program ADC sensor LDR ialah:

```

.....
// Program : Get ADC Value from LDR Sensor

#include "mbed.h"

#define Pin_LDR1 PC_4
#define Pin_LDR2 PC_5

AnalogIn LDR1_ADC(Pin_LDR1);
AnalogIn LDR2_ADC(Pin_LDR2);

```

```

float LDR1, LDR2, Aver_LDR1, Aver_LDR2;

void sensor_LDR();

int main(){
    sensor_LDR();
}

void sensor_LDR(){
    float t_ldr1, t_ldr2;
    for(int a=0; a<77; a++){
        LDR1 = LDR1_ADC.read();
        LDR2 = LDR2_ADC.read();
        t_ldr1 = t_ldr1 + LDR1;
        t_ldr2 = t_ldr2 + LDR2;
    }
    Aver_LDR1 = t_ldr1 / 77;
    Aver_LDR2 = t_ldr2 / 77;
}

```

Listing code diatas dibuat berdasarkan atas flowchart pada gambar 3.11. Dimana berdasar atas flowchart tersebut, langkah awal dalam mendapatkan nilai ADC dari sensor LDR ialah dengan memasukkan *file header* yang akan dipakai dalam program. *File Header* atau sering disebut juga *Library* merupakan sekumpulan pendeklarasian fungsi yang dibuat untuk memenuhi tugas tertentu. *file header* merupakan bagian kode program yang wajib ada. Berdasarkan sumbernya *file header* dibagi menjadi dua macam, yaitu internal dan eksternal. *File header* internal ialah *file header* yang sudah tersedia dalam satu paket instalasi dengan kompilator ataupun IDE-nya. Dalam program ini, contoh *file header* internal ialah `stdio.h`. fungsi yang dapat digunakan dengan memanggil *file header* tersebut contohnya ialah `printf` yang dapat digunakan sebagai pencetak string ke *interface*. Sedangkan untuk *file header* eksternal sendiri ialah kebalikan dari *file header* internal yaitu *file*

header belum tersedia dalam satu paket instalasi dengan kompilator ataupun IDE-nya. Sehingga perlunya memasukkan *file header* eksternal agar fungsi-fungsi nya dapat digunakan dalam sebuah program. Terdapat dua *file header* eksternal dalam program ini yaitu *mbed.h* dan *rtos.h*. untuk *file header* *mbed.h*, di dalam *header* tersebutlah fungsi *pheripheral* seperti ADC, PWM, USART dideklarasikan sehingga dapat digunakan atau mengaksesnya pada program utama yaitu *main.c*. misalnya pada program sebelumnya yang mana *pheripheral* yang butuh diakses atau ingin digunakan ialah ADC. Maka cara yang digunakan untuk mengaksesnya ialah dengan menginisialisasikan pin *pheripheral* yang bersangkutan lalu kemudian tinggal menggunakan fungsinya. Selanjutnya untuk *file header* *rtos.h* sudah dijelaskan pada subbab sebelumnya yang intinya ia bertugas sebagai pembuat jalur untuk *multitasking* dengan membuat *thread* yang diinginkan. Setelah *header* sudah dimasukkan, maka selanjutnya ialah mendefinisikan pin yang akan digunakan untuk sensor LDR, dalam *listing code* tersebut ialah pin PC_4 sebagai Pin_LDR1 dan PC_5 sebagai Pin_LDR2. Lalu berikutnya masuk pada pendeklarasian pin yang akan digunakan sebelumnya pada *pinmap* sehingga *pheripheral* yang terdapat pada pin tersebut dapat digunakan yaitu dengan memanggil fungsi *AnalogIn*, inilah yang dimaksud inisialisasi *pheriper*al sebelumnya. Jika sudah maka selanjutnya ialah menggunakan *peripheral* ADC pada pin sebelumnya dengan menggunakan fungsi *.read()*, fungsi tersebutlah yang bertugas untuk membaca nilai *Analog* yang masuk dari sensor kemudian merubahnya ke dalam bentuk digital agar mudah dibaca dan dipahami *user*. Maka program untuk mendapatkan nilai ADC dari sensor LDR sudah selesai. Namun untuk kasus ini, ditambahkan blok *listing code* agar pembacaan ADC dilakukan tiap sensor LDR mendapatkan nilai setiap 77 kali terlebih dahulu

lalu dicari rata-ratanya. Hal ini dilakukan agar didapatkannya data yang lebih akurat berdasarkan hasil rata-rata tersebut. Tujuan lain dari perlakuan tersebut ialah agar tidak adanya data nilai sensor LDR yang tergeser dan digantikan oleh nilai sebelum-sebelumnya sehingga pergerakan respon PV akan menjadi lebih halus

Setelah nilai ADC tiap LDR ditemukan yang mana pada kasus ini nilai disimpan pada variabel `Aver_LDR1` dan `Aver_LDR2`, maka selanjutnya nilai tersebut akan diolah pada blok program algoritma ANFIS yang terdiri dari tiga thread, yaitu NN (neural network), inferensi rule dan defuzifikasi. Berikut ini ialah sampel *listing code thread* yang pertama yaitu NN.

```

.....
// Program : Thread::NN

#include "mbed.h"
#include "rtos.h"

Thread thread();

void thread_NN() {

    float x1, x2, x1_norm, x2_norm;
    float Z_inj, Zj, voj, vij1, vij2;
    float Y_ink, Yk, wok, wjk;
    float out_denorm;
    float in1, in2;

    in1 = LDR1_aver;
    in2 = LDR2_aver;

    //decision block
    if(in1 <= 350) {
        if(in1 <= 300) {
            if(in2 <= 325) {
                voj=0.93; vij1=0.95; vij2=0.95;
            }
        }
    }
}

```

```

        if(in2 <= 310) {
            wok=0.24; wjk=-0.18;
        }
        else if(in2 > 310  && in2 < 320) {
            wok=0.35; wjk=0.21;
        }
        else {
            wok=0.48; wjk=0.42;
        }
    }
    else if(in2 > 325 && in2 < 350) {
        voj=0.97; vij1=0.97; vij2=0.97;
        if(in2 < 340) {
            wok=0.7; wjk=0.8;
        }
        else {
            wok=1.2; wjk=1.1;
        }
    }
    else {
        voj=1.05; vij1=0.99; vij2=0.99;
        wok=1.3; wjk=1.4;
    }
}

x1 = LDR1_aver;
x2 = LDR2_aver;

x1_norm = (((x1-0)/(1000-0))*((1-0)+(0)));
x2_norm = (((x2-0)/(1000-0))*((1-0)+(0)));

Z_inj = voj + ((x1_norm*vij1)+(x2_norm*vij2));
Zj = 1/(1+(exp(-Z_inj)));

Y_ink = wok + ((Zj*wjk)+(Zj*wjk));
Yk = 1/(1+(exp(-Y_ink)));

out_temp = Yk;

```

.....

Listing code tersebut berdasarkan atas *flowchart* pada gambar 3.12, dapat diketahui dari *flowchart* tersebut langkah awal dari pembuatan *thread* inilah deklarasi variabel. Dalam kasus ini variabel yang digunakan antara lain: x_1 , x_2 hingga in_1 dan in_2 yang bertipe data *float*. Jika sudah terdeklarasi maka selanjutnya nilai ADC dari LDR1 dan LDR2 akan dimasukkan pada variabel lain misalnya in_1 dan in_2 . Kemudian variabel in_1 in_2 yang berisikan nilai ADC kedua sensor selanjutnya diproses pada *decision block* untuk dikondisikan, maksud dari dikondisikan disini ialah in_1 dan in_2 diberikan keadaan, yang mana jika keadaan tersebut terpenuhi maka nilai bobot dan bias akan didapatkan. Terdapat banyak keadaan untuk mengklasifikasikan nilai in_1 dan in_2 sehingga nilai bobot dan bias yang akan didapatkan juga berbeda-beda tergantung klasifikasi atau keputusan mana yang akan diambil oleh algoritma ketika diberi input. Nilai bobot dan bias yang tertera pada *listing code* ialah nilai yang cenderung muncul atau dalam matematika ialah modus pada beberapa *range* data. Jika nilai bobot dan bias sudah didapatkan maka selanjutnya ialah menghitung nilai output untuk input yang dimaksudkan. Adapun langkah-langkahnya sesuai dengan algoritma JST bagian *FeedForward* yaitu dengan mencari nilai unit tersembunyi(Z_j) terlebih dahulu dengan mencari nilai aktivasi untuk sambungan tersebut. setelah itu baru mencari nilai unit output(Y_k) sesuai dengan fungsi aktivasinya juga. Pada proses ini nilai bobot dan bias akan secara otomatis masuk pada formula yang ditulis sehingga nilai Y_k pun akan secara langsung didapatkan. Namun perlu diketahui jikalau *range* yang digunakan pada input dan output tergolong besar sehingga perlunya untuk menormalisasi nilai input dan output untuk dijadikan *range* 0 hingga 1. Maka program `thread::NN` selesai dengan menyimpan


```

    checker1 = 5;
}

if(in2 <= 350) {
    checker2 = 1;
}
if(in2 >= 300 && in2 <=500) {
    checker2 = 2;
}
if(in2 >= 450 && in2 <= 550) {
    checker2 = 3;
}
if(in2 >= 500 && in2 <= 700) {
    checker2 = 4;
}
if(in2 >= 650) {
    checker2 = 5;
}

if (checker1 == checker2) {
    out_f = 0;
    m1 = out_f;
    m2 = out_f;
}
else if(checker1 > checker2) {
    if (out_f < 0) {
        min = out_f;
        plus = out_f;
        m1 = min;
        m2 = positive(plus);
    }
    else if (out_f > 0) {
        min = out_f;
        plus = out_f;
        m1 = negative(min);
        m2 = plus;
    }
}
else if(checker1 < checker2) {

```

```

if (out_f > 0) {
    plus = out_f;
    min = out_f;
    m1 = plus;
    m2 = negative(min);
}
else if (out_f < 0) {
    plus = out_f;
    min = out_f;
    m1 = positive(plus);
    m2 = min;
}
}

```

.....

Listing code tersebut juga berdasarkan atas *flowchart* sebelumnya yaitu pada gambar 3.13, berdasarkan *flowchart* tersebut, langkah awal pembuatan *thread* ini sama dengan program sebelum-sebelumnya yaitu deklarasi variabel yaitu in1, in2 bertipekan data *float* hingga checker1 dan checker2 yang bertipekan data *integer*. Selanjutnya ialah memasukkan nilai ADC yang diperoleh dari program get_ADC kedalam variabel lain yaitu in1 dan in2. Setelah in1 dan in2 berisikan sebuah nilai maka nilai tersebut dilakukan pengecekan atau pengklasifikasian seperti pada program thread:NN namun bedanya disini yang didapatkan ialah nilai checker1 dan checker2 antara nilai satu hingga lima, nilai tersebut mengidikasikan himpunan fuzzy dari VS(*Very Small*) hingga VB(*Very Big*) yang sudah dijelaskan pada subbab sebelumnya. Dari nilai tersebut maka nilai in1 dan in2 dapat diklasifikasikan pada *Rule* mana yang aktif atau yang dipakai. Jika sudah diketahui *rule* mana yang aktif maka selanjutnya ialah menyesuaikan nilai variabel out_f pada *rule* yang aktif. Cara menyesuaikan ialah melakukan cek pada nilai tersebut, jika sesuai maka nilai akan diarahkan ke variabel lain, jika tidak maka

```
// Program : Thread::Defuzifikasi

#include "mbed.h"
#include "rtos.h"

Thread thread;

void thread_defuzifikasi() {
    float m1_denorm, m2_denorm;

    m1_denorm = ((m1-(0))/(1-(0)))*(93,43-(-93,43))+(-
        93,43);
    m2_denorm = ((m2-(0))/(1-(0)))*(93,43-(-93,43))+(-
        93,43);
}
```

Listing code diatas ialah *thread* ketiga atau yang terkakhir dan sesuai dengan *flowchart* pada gambar 3.14. *Thread* ini bertugas untuk mendenormalisasi nilai output dari variabel m1 dan m2 yang memiliki 0 hingga 1 kedalam variabel m1_denorm dan m2_denorm yang memiliki nilai dengan *range output* yaitu -93,4 hingga 93,4. Nilai-nilai tersebutlah yang akan diproses oleh PWM untuk dapat menggerakkan motor

4.3 Hasil Eksperimen Algoritma ANFIS

Pada subbab sebelumnya yaitu 3.3 Eksperimen Algoritma ANFIS sudah dijelaskan metodologi eksperimen terhadap algoritma ANFIS. Pada tabel 4.2 hingga 4.4 akan disajikan hasil eksperimen Algoritma ANFIS.

Tabel 4.2 Hasil Eksperimen 1: Flaslight pada LDR1

Input1	Input2	Output1	Output2
956	103	-79,7	79,7
912	103	-79,7	79,7
887	130	-79,7	79,7
883	136	-79,7	79,7
840	142	-79,7	79,7
827	168	-79,7	79,7
803	168	-79,7	79,7
793	172	-79,7	79,7
776	174	-79,7	79,7
765	181	-79,7	79,7

Tabel 4.3 Hasil Eksperimen 2: *Flashlight* antara LDR1 dan LDR2

Input1	Input2	Output1	Output2
523	503	-29,2	29,2
501	487	-31,15	31,15
526	513	-12,18	12,18
516	508	-15,15	15,15
529	508	-25,08	25,08
503	511	17,21	-17,21
507	504	-8,59	8,59
503	501	-0,13	0,13
497	505	22,31	-22,31
503	508	4,59	-4,59

Tabel 4.4 Hasil Eksperimen 3: *Flashlight* pada LDR2

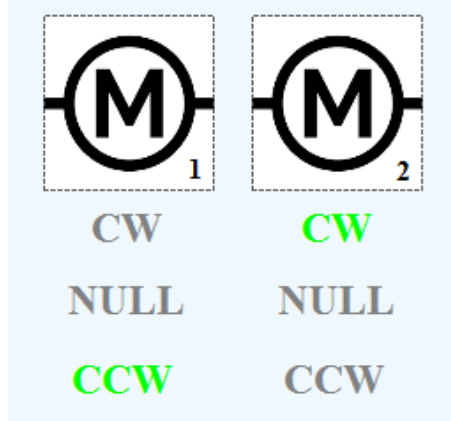
Input1	Input2	Output1	Output2
881	134	79,7	-79,7
888	138	79,7	-79,7
867	132	79,7	-79,7
838	154	79,7	-79,7
823	173	79,7	-79,7
816	181	79,7	-79,7
798	186	79,7	-79,7
778	187	79,7	-79,7
775	183	79,7	-79,7
757	188	79,7	-79,7

Data pada Tabel-tabel diatas yaitu Tabel 4.2 hingga 4.4 ialah sampel data yang dimabil dari eksperimen yang dilakukan yang

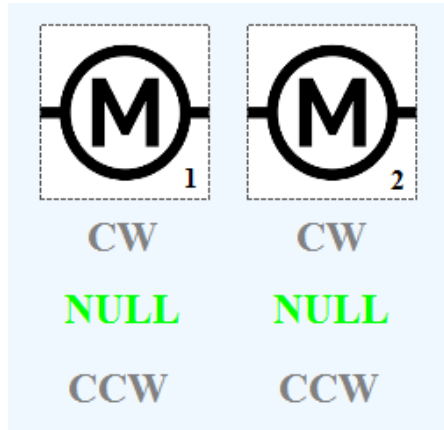
diambil dari program OOP yang telah dibuat. Data tersebut hanya meliputi nilai pada Input 1 dan Input 2 saat itu hingga nilai Output 1 dan Output 2 nya. Misal pada Tabel 4.2 ialah hasil dari eksperimen pertama yang mana *flashlight* hanya diarahkan pada LDR1, dan didapatkanlah hasil perputaran maksimum dari motor1 ataupun motor2, hal tersebut dapat dilihat dari nilai untuk Output1 dan Output2. Dengan data eksperimen berupa tabel tersebut, dapat dikatakan jika Algoritma ANFIS berhasil dirancang dengan memodifikasinya. Pernyataan tersebut dibenarkan karena pada saat Eksperimen kedua yaitu *Flashlight* ditaruh diantara sensor LDR1 dan LDR2, nilai pada variabel Output1 dan Output2 mendekati Nol atau Null, yang berarti dalam realisasi nyatanya, LDR1 dan LDR2 akan mengarahkan motor ke intensitas matahari terbanyak.

Selain hasil tabel sebelumnya, juga didapatkan hasil dari program OOP berupa indikasi arah perputaran motor1 dan motor2 ketika ketiga eksperimen tersebut dilakukan. Pada Gambar 4.1 hingga 4.3 akan disajikan gambar yang menjelaskan indikasi arah perputaran kedua motor pada program OOP yang telah dibuat.

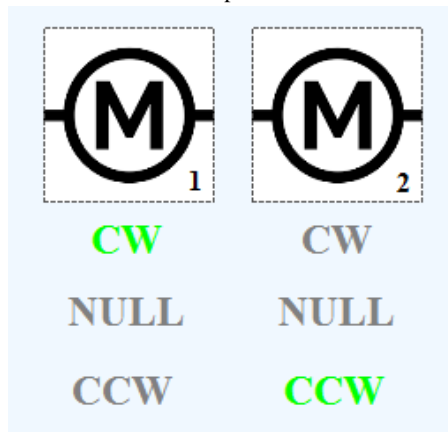
Gambar 4.1 Hasil Eksperimen 1: LDR1>LDR2



Gambar 4.2 Hasil Eksperimen 1: $LDR1=LDR2$



Gambar 4.3 Hasil Eksperimen 1: $LDR1 < LDR2$



Sudah dijelaskan sebelumnya jika pada Gambar-gambar diatas yaitu Gambar 4.1 hingga 4.3 menjelaskan mengenai arah perputaran kedua motor, berarti terdapat tiga kondisi yang berlaku

untuk masing-masing output, yaitu CW, Null, dan CCW. Pada Gambar 4.1 motor 1 bergerak kekiri atau CCW hal tersebut ditandai dengan berubahnya warna font dari *grey*(abu-abu) atau non aktif menjadi warna *lime green*(hijau lemon) atau aktif untuk kasus ini, sedangkan untuk motor 2 bergerak sebaliknya yaitu kekanan atau CW, hal ini dapat terjadi dikarenakan nilai LDR1 lebih besar daripada LDR2. Hal tersebut berlaku sebaliknya untuk Gambar 4.2 dimana nilai LDR1 lebih kecil daripada LDR2, sehingga motor 1 bergerak kekanan atau CW, sedangkan motor 2 bergerak ke kiri atau CCW. Lain halnya juga ketika nilai LDR1 dan LDR2 sama. Maka motor 1 dan motor 2 tidak akan berputar atau dalam keadaan Null, seperti pada Gambar 4.3. Dengan hasil eksperimen dan simulasi berupa program OOP tersebut, dapat dikatakan bahwa Algoritma berhasil dirancang sesuai keinginan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi algoritma kecerdasan buatan modifikasi pada dual-axis solar tracking system dengan desain vertical-axis array berbasis STM32, didapatkan beberapa kesimpulan sebagai berikut:

1. Telah dihasilkan divais yang mampu menjalankan algoritma kecerdasan buatan modifikasi untuk dual-axis solar tracking system dengan desain vertical-axis array berbasis STM32 sebagai salah satu sumber energi pada *smart classroom* berdasarkan atas hasil eksperimen pada tabel 4.2 hingga tabel 4.4
2. Telah dihasilkan divais yang mampu meningkatkan daya keluaran sistem fotovoltaik pada *dual-axis solar tracking system* dengan desain *vertical-axis array* berdasarkan atas hasil eksperimen pada Gambar 4.1 hingga Gambar 4.3

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, didapatkan saran yang mana diperlukannya pengembangan algoritma dengan penambahan variabel daya untuk mengoptimalkan kinerja dari *dual-axis solar tracking system* dengan desain *vertical-axis array*.

“Halaman ini sengaja dikosongkan”

DAFTAR PUSTAKA

- [1] I. Winarno dan F. Wulandari. “*SOLAR TRACKING SYSTEM SINGLE AXIS PADA SOLAR SEL UNTUK MENGOPTIMALKAN DAYA DENGAN METODE ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)*”. dalam *Seminar Nasional Sains dan Teknologi*, Fakultas Teknik Universitas Muhammadiyah Jakarta, Surabaya: Teknik Elektro Universitas Hang Tuah, 2017.
- [2] M, K., Pebriningtyas, Musyafa, A., & Indriawati, K. “Penelusuran Daya Maksimum Pada Panel Photovoltaic Menggunakan Kontrol Logika Fuzzy Di Kota Surabaya”. 2013.
- [3] S, R., S, G., & K, M. “Development of FPGA-based Dual-Axis Solar Tracking System”. Volume 2 No. 4. 2013.
- [4] Budi, S. H. “PENGEMBANGAN SOLAR TRACKER SINGLE AXIS BERBASIS ARDUINO UNTUK MENINGKATKAN PEROLEHAN ENERGI MATAHARI”. 2015.
- [5] Widodo, & T, R. “*Solar Cell: Sumber Energi Masa Depan Yang Ramah Lingkungan*”. Surabaya: Politeknik Elektronika Negeri Surabaya - ITS. 2003.
- [6] Basil M Hamed, M. S.-M. “*Fuzzy Controller Design Using FPGA for Photovoltaic Maximum Power Point Tracking*”. Palestine: (IJARAI) International Journal of Advanced Research in Artificial Intelligence. 2012.
- [7] Y. J. Huang, “The Design and Implementation of a Solar Tracking Generating Power System,” *Engineering Letters*, vol. 4, p. 17, 2009.
- [8] Trishan E., C. P. “*Comparison of Photovoltaic Array Maximum Power Point Trackin Techniques*”. 35th Annual IEEE Power Electron. 2007.

- [9] T, T., & L, K. “*Design of a Solar Tracker System for PV Power Plants*”. Romania: Electrical Engineering Faculty, University Polithenica of Bucharest. 2010.
- [10] S, I. W., & A, K. U. “*SOLAR TRACKER CERDAS DAN MURAH BERBASIS MIKROKONTROLER 8 BIT ATmega8535*”. Singaraja: Jurusan Teknik Elektronika, Universitas Pendidikan Ganesha. 2016.
- [11] STMicroelectronics. “Microcontrollers”. stmicroelectronics. Tersedia dari <http://www.st.com/en/microcontrollers.html>. (diakses 19 Februari 2018)
- [12] STMicroelectronics. “STM32 32-bit ARM Cortex MCUs”. stmicroelectronics. Tersedia dari http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus.html (diakses 19 Februari 2018)
- [13] K. DIAW A/L EH SUT. “FPGA BASED MAXIMUM POWER POINT TRACKING OF PHOTOVOLTAIC SYSTEM”. Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein onn Malaysia. 2014
- [14] Muhammad , Amar dkk. “Rancang Bangun Sistem Penjejak Matahari 2 Sumbu Berbasis Kontrol *Adaptive Neuro-Fuzzy Inference System* (ANFIS)”. Surabaya: Institut Teknologi Sepuluh Nopember. 2012.
- [15] F.U. Azmi. “Desain Penggerak Panel Surya Dual Axis pada Smartclassroom”. Surabaya: Departemen Teknik Instrumentasi Fakultas Vokasi, Institut Teknologi Sepuluh Nopember. 2018.
- [16] K.K. Saputro. “Rancang Bangun control motor DC Stepper Berdesain Dual-Axis Vertical Array Sebagai Sumber Tenaga Pada Smartclassroom”. Surabaya: Departemen

Teknik Instrumentasi Fakultas Vokasi, Institut Teknologi Sepuluh Nopember. 2018.

- [17] HAW. Kesuma. “Perancangan Sistem Panel Surya Sebagai Sumber Energi Pada Smart Classroom”. Surabaya: Departemen Teknik Instrumentasi Fakultas Vokasi, Institut Teknologi Sepuluh Nopember. 2018.
- [18] E.Q. Nugraha. “Perancangan sistem monitoring tracking photovoltaic sebagai sumber energi pada smartclassroom berbasis wireless”. Surabaya: Departemen Teknik Instrumentasi Fakultas Vokasi, Institut Teknologi Sepuluh Nopember. 2018.

“Halaman ini sengaja dikosongkan”

LAMPIRAN A

(Data Z Fuzzy(Tk), Beban(Vij dan Wkj) dan Bias(Voj dan Wok) Neural Network, Error(E) ANFIS, Final Output ANFIS(Yk))

Input1	Input2	E	epoch	vij1	vij2	voj	wkj	wok	Yk	Tk
300	300	-0,005	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,741773	0
300	301	-0,0025	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,740565	1,87
300	302	8,57E-07	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,739358	3,74
300	303	0,0025	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,738152	5,61
300	304	0,004986	1	0,951036	0,95154	0,931764	-0,18279	0,229651	3,736947	7,47
300	305	0,004999	217	0,950713	0,951211	0,930762	-0,14681	0,239969	5,590187	9,34
300	306	0,004993	216	0,950462	0,950955	0,929982	-0,11101	0,250695	7,450846	11,207
300	307	0,004999	213	0,950282	0,950771	0,929426	-0,07573	0,261736	9,300455	13,074
300	308	0,004993	212	0,950171	0,950657	0,929082	-0,04064	0,273197	11,15488	14,941
300	309	0,004992	210	0,950127	0,950612	0,928945	-0,00593	0,285028	13,00345	16,808
300	310	0,004993	208	0,950148	0,950633	0,929009	0,028426	0,297239	14,84602	18,675
300	311	0,004999	206	0,950232	0,95072	0,929269	0,062438	0,30984	16,68246	20,542
300	312	0,004999	205	0,950377	0,950871	0,929719	0,096261	0,322899	18,51998	22,409

300	313	0,004995	204	0,950583	0,951086	0,930357	0,12987	0,336416	20,35612	24,276
300	314	0,004991	203	0,950847	0,951362	0,931177	0,163252	0,350397	22,18914	26,143
300	315	0,004998	201	0,951167	0,951699	0,932172	0,19627	0,364791	24,01033	28,01
300	316	0,004993	201	0,951545	0,952097	0,933346	0,22924	0,379749	25,83604	29,877
300	317	0,004991	200	0,951977	0,952553	0,934692	0,261979	0,395201	27,65495	31,744
300	318	0,004993	199	0,952463	0,953068	0,936206	0,294502	0,411166	29,46667	33,611
300	319	0,004998	198	0,953	0,953639	0,937884	0,326824	0,427663	31,2708	35,478
300	320	0,004997	198	0,953591	0,95427	0,939733	0,359096	0,444785	33,07451	37,345
300	321	0,004994	198	0,954234	0,954958	0,94175	0,391292	0,46254	34,87505	39,212
300	322	0,004999	197	0,954925	0,9557	0,943923	0,423265	0,480861	36,6628	41,079
300	323	0,004992	198	0,955671	0,956502	0,946271	0,455323	0,499947	38,45361	42,946
300	324	0,004997	197	0,956462	0,957357	0,94877	0,487145	0,519627	40,22809	44,813
300	325	0,004993	198	0,957306	0,958271	0,951443	0,519053	0,540124	42,00264	46,68
300	326	0,004993	198	0,958198	0,959241	0,954276	0,550873	0,561353	43,76596	48,547
300	327	0,004996	198	0,959137	0,960264	0,957269	0,58262	0,583348	45,51736	50,414
300	328	0,004995	199	0,960128	0,961348	0,960435	0,614447	0,606244	47,26349	52,281
300	329	0,004999	199	0,961165	0,962484	0,963759	0,646196	0,629959	48,99392	54,148
300	330	0,004993	201	0,962257	0,963685	0,967274	0,678166	0,654754	50,72298	56,015

300	331	0,004996	201	0,963393	0,964939	0,970943	0,710033	0,680417	52,43128	57,882
300	332	0,004998	202	0,964578	0,966251	0,974787	0,741974	0,707124	54,12632	59,749
300	333	0,004993	204	0,965818	0,967627	0,978824	0,774122	0,735033	55,81293	61,616
300	334	0,004998	204	0,9671	0,969054	0,983016	0,806169	0,763923	57,47292	63,483
300	335	0,004997	206	0,968436	0,970547	0,987405	0,838434	0,794125	59,12059	65,35
300	336	0,004999	207	0,96982	0,972097	0,99197	0,870749	0,825535	60,74501	67,217
300	337	0,004996	209	0,971257	0,973711	0,996733	0,903262	0,858355	62,35118	69,084
300	338	0,004996	210	0,97274	0,975382	1,001673	0,93581	0,892479	63,92881	70,951
300	339	0,004998	211	0,974269	0,977109	1,00679	0,968406	0,927971	65,47627	72,818
300	340	0,004996	213	0,97585	0,978901	1,012109	1,001194	0,965049	66,99798	74,685
300	341	0,004999	213	0,977468	0,98074	1,017582	1,033869	1,003427	68,47782	76,552
300	342	0,004997	215	0,979136	0,982642	1,023258	1,066735	1,043519	69,92751	78,419
300	343	0,004997	215	0,98084	0,98459	1,029084	1,099479	1,085	71,3313	80,286
300	344	0,004998	215	0,982577	0,986581	1,035059	1,132112	1,127925	72,6885	82,153
300	345	0,004997	215	0,984346	0,988616	1,041181	1,164633	1,172334	73,99799	84,02
300	346	0,004997	214	0,986137	0,990682	1,047417	1,196892	1,218048	75,25306	85,887
300	347	0,004999	212	0,987941	0,992769	1,053735	1,228747	1,264877	76,44848	87,754
300	348	0,004999	210	0,989756	0,994874	1,060129	1,260204	1,312822	77,58513	89,621

300	349	0,004998	207	0,991571	0,996985	1,066561	1,291116	1,361639	78,65883	91,488
300	350	0,004998	203	0,993374	0,999089	1,072992	1,321339	1,411057	79,66664	93,355
300	351	0,004997	1	0,993374	0,999089	1,072992	1,321339	1,411057	79,6688	93,355
301	300	-0,005	2012	0,949891	0,95575	0,928649	-0,21732	0,217717	1,868455	-1,87
301	301	-0,0025	1	0,949891	0,95575	0,928649	-0,21732	0,217717	1,86701	0
301	302	-8,8E-05	1	0,949891	0,95575	0,928649	-0,21732	0,217717	1,865566	1,8
301	303	0,002308	1	0,949891	0,95575	0,928649	-0,21732	0,217717	1,864123	3,59
301	304	0,004717	1	0,949891	0,95575	0,928649	-0,21732	0,217717	1,862681	5,39
301	305	0,004991	194	0,949546	0,955401	0,927585	-0,18603	0,2262	3,454143	7,19
301	306	0,004995	211	0,949226	0,955075	0,926599	-0,15133	0,235998	5,234721	8,98
301	307	0,004995	210	0,948972	0,954817	0,925819	-0,11677	0,246177	7,023429	10,778
301	308	0,004993	208	0,948786	0,954626	0,925245	-0,08257	0,256682	8,808324	12,574
301	309	0,004994	206	0,948664	0,954501	0,924872	-0,04872	0,267519	10,58886	14,37
301	310	0,004997	204	0,948606	0,95444	0,924691	-0,01521	0,278696	12,36462	16,166
301	311	0,004993	203	0,948608	0,954443	0,924699	0,018097	0,29027	14,14253	17,962
301	312	0,004992	202	0,94867	0,954508	0,924891	0,051048	0,302241	15,91512	19,758
301	313	0,004991	200	0,948791	0,954633	0,92526	0,083791	0,314568	17,68503	21,554
301	314	0,004997	198	0,948967	0,954816	0,925801	0,116187	0,327257	19,44608	23,35

301	315	0,004993	199	0,9492	0,955061	0,92652	0,148386	0,340507	21,21143	25,146
301	316	0,004996	196	0,949486	0,95536	0,927399	0,180382	0,354059	22,9679	26,942
301	317	0,004995	196	0,949825	0,955718	0,928446	0,212175	0,368132	24,72371	28,738
301	318	0,004995	195	0,950217	0,956131	0,929654	0,243756	0,382664	26,47442	30,534
301	319	0,004992	194	0,950658	0,956599	0,931018	0,275291	0,397651	28,22411	32,33
301	320	0,004993	193	0,951148	0,957121	0,932536	0,30661	0,413109	29,96595	34,126
301	321	0,004998	192	0,951687	0,957695	0,934205	0,337733	0,429057	31,69993	35,922
301	322	0,004995	193	0,952278	0,958327	0,936041	0,368802	0,445673	33,43648	37,718
301	323	0,004998	192	0,952914	0,95901	0,938023	0,399649	0,462804	35,16181	39,514
301	324	0,004999	192	0,953599	0,959747	0,940159	0,430462	0,480539	36,88405	41,31
301	325	0,004993	192	0,95433	0,960537	0,942446	0,461352	0,498904	38,60558	43,106
301	326	0,004998	191	0,955104	0,961375	0,944872	0,492017	0,517816	40,31159	44,902
301	327	0,004994	192	0,955928	0,962269	0,947461	0,522777	0,537493	42,01845	46,698
301	328	0,004994	192	0,956796	0,963215	0,950198	0,553456	0,557849	43,71501	48,494
301	329	0,004997	192	0,957708	0,964213	0,953082	0,584071	0,578913	45,40066	50,29
301	330	0,004995	193	0,958669	0,965266	0,956128	0,614769	0,600817	47,0821	52,086
301	331	0,004998	193	0,959672	0,966369	0,959321	0,645394	0,623474	48,74907	53,882
301	332	0,004994	195	0,960728	0,967534	0,962692	0,676091	0,647149	50,41223	55,678

301	333	0,004994	196	0,96183	0,968753	0,966224	0,706708	0,671742	52,06092	57,474
301	334	0,005	195	0,962968	0,970016	0,969883	0,737402	0,697064	53,6906	59,27
301	335	0,004997	197	0,964158	0,97134	0,973725	0,768331	0,72352	55,31531	61,066
301	336	0,004999	198	0,965393	0,972719	0,97773	0,799173	0,751006	56,92031	62,862
301	337	0,004998	199	0,966674	0,974153	0,981899	0,830233	0,779575	58,51149	64,658
301	338	0,004998	201	0,968006	0,975649	0,986253	0,861205	0,809402	60,08329	66,454
301	339	0,004995	202	0,969382	0,977198	0,990771	0,892527	0,840392	61,63955	68,25
301	340	0,004999	202	0,970795	0,978794	0,995432	0,923753	0,872433	63,16402	70,046
301	341	0,004995	205	0,972265	0,98046	1,000307	0,955182	0,906055	64,67372	71,842
301	342	0,004998	205	0,973771	0,98217	1,005323	0,986516	0,940808	66,14758	73,638
301	343	0,004996	207	0,975326	0,983943	1,01053	1,01805	0,977096	67,59812	75,434
301	344	0,004996	207	0,976915	0,985758	1,015878	1,049631	1,014603	69,01182	77,23
301	345	0,005	207	0,978537	0,987618	1,021368	1,081121	1,053388	70,38519	79,026
301	346	0,004999	208	0,980199	0,989529	1,027025	1,11267	1,093681	71,72326	80,822
301	347	0,004997	209	0,981901	0,99149	1,032846	1,144109	1,135517	73,02136	82,618
301	348	0,004996	208	0,983623	0,993482	1,038773	1,175287	1,178539	74,26882	84,414
301	349	0,004996	207	0,985365	0,995502	1,044804	1,206226	1,222767	75,46587	86,21
301	350	0,005	533	0,990638	1,001633	1,063266	1,296906	1,361163	78,71907	91,6

301	500	0,004999	135	0,991694	1,003387	1,067002	1,316916	1,392984	79,68918	93,4
302	300	-0,005	1990	0,945388	0,969459	0,915285	-0,24962	0,203672	-0,00467	-3,74
302	301	-0,0024	1	0,945388	0,969459	0,915285	-0,24962	0,203672	-0,00637	-1,8
302	302	1,08E-05	1	0,945388	0,969459	0,915285	-0,24962	0,203672	-0,00806	0
302	303	0,002328	1	0,945388	0,969459	0,915285	-0,24962	0,203672	-0,00975	1,73
302	304	0,004644	1	0,945388	0,969459	0,915285	-0,24962	0,203672	-0,01144	3,46
302	305	0,004998	184	0,945007	0,969075	0,914133	-0,22035	0,2111	1,453025	5,19
302	306	0,004993	210	0,944622	0,968684	0,912966	-0,18606	0,22015	3,183904	6,92
302	307	0,004997	207	0,944305	0,968363	0,912011	-0,15227	0,229448	4,904567	8,65
302	308	0,004991	206	0,944053	0,968106	0,91125	-0,11866	0,239085	6,630424	10,38
302	309	0,004998	203	0,943866	0,967914	0,910685	-0,08555	0,248974	8,344391	12,11
302	310	0,004995	202	0,94374	0,967785	0,910306	-0,05261	0,259217	10,06258	13,84
302	311	0,004997	200	0,943675	0,967717	0,910108	-0,02002	0,269765	11,77506	15,57
302	312	0,004992	199	0,943668	0,96771	0,910087	0,012375	0,280673	13,48912	17,3
302	313	0,004994	197	0,943718	0,967762	0,910238	0,044416	0,291894	15,1955	19,03
302	314	0,004993	196	0,943824	0,967872	0,910557	0,076268	0,303491	16,9021	20,76
302	315	0,004998	194	0,943983	0,968038	0,911038	0,107778	0,315413	18,59992	22,49
302	316	0,004991	194	0,944197	0,968262	0,911682	0,13925	0,327784	20,30449	24,22

302	317	0,004995	192	0,944461	0,968539	0,91248	0,170359	0,340486	21,99737	25,95
302	318	0,004998	191	0,944776	0,968871	0,913432	0,201289	0,353597	23,68762	27,68
302	319	0,004992	191	0,945142	0,969257	0,91454	0,23217	0,367186	25,38155	29,41
302	320	0,004999	189	0,945554	0,969694	0,915789	0,262695	0,381124	27,0613	31,14
302	321	0,004998	189	0,946016	0,970185	0,917191	0,293191	0,395572	28,74395	32,87
302	322	0,004991	189	0,946526	0,970729	0,918741	0,323622	0,410526	30,42659	34,6
302	323	0,004992	188	0,947081	0,971322	0,920431	0,353832	0,425923	32,0996	36,33
302	324	0,004998	187	0,94768	0,971965	0,922258	0,38385	0,441785	33,76352	38,06
302	325	0,004999	187	0,948325	0,972659	0,92423	0,413834	0,45821	35,42612	39,79
302	326	0,004998	187	0,949015	0,973404	0,926345	0,443768	0,475204	37,0853	41,52
302	327	0,004997	187	0,94975	0,9742	0,928602	0,473641	0,492782	38,73945	43,25
302	328	0,004996	187	0,950529	0,975046	0,930999	0,503452	0,510958	40,38726	44,98
302	329	0,004997	187	0,95135	0,975941	0,933536	0,533202	0,529753	42,02766	46,71
302	330	0,004999	187	0,952215	0,976885	0,936211	0,562898	0,549187	43,65967	48,44
302	331	0,004995	188	0,953125	0,977883	0,939039	0,592681	0,56938	45,28983	50,17
302	332	0,004995	188	0,954077	0,97893	0,942002	0,62239	0,590245	46,90785	51,9
302	333	0,005	188	0,955069	0,980024	0,945103	0,652042	0,611816	48,51344	53,63
302	334	0,005	189	0,956107	0,981172	0,948356	0,681792	0,63423	50,11341	55,36

302	335	0,004998	190	0,95719	0,982373	0,951762	0,711621	0,657508	51,70516	57,09
302	336	0,004995	191	0,958317	0,983626	0,95532	0,741518	0,681674	53,28652	58,82
302	337	0,004999	191	0,959482	0,984926	0,959012	0,771342	0,706642	54,84845	60,55
302	338	0,004995	193	0,960697	0,986286	0,962878	0,801397	0,732703	56,40516	62,28
302	339	0,004999	193	0,961948	0,987691	0,966878	0,83137	0,759625	57,93876	64,01
302	340	0,004996	195	0,96325	0,989156	0,971053	0,861571	0,787723	59,46324	65,74
302	341	0,004995	196	0,964593	0,990673	0,975383	0,89183	0,816886	60,96808	67,47
302	342	0,004995	197	0,965979	0,992243	0,979868	0,922152	0,84716	62,45175	69,2
302	343	0,004995	198	0,967406	0,993864	0,98451	0,95254	0,878592	63,91263	70,93
302	344	0,004996	199	0,968875	0,995536	0,989309	0,982996	0,911227	65,34897	72,66
302	345	0,004997	200	0,970384	0,997261	0,994266	1,01352	0,945111	66,75897	74,39
302	346	0,004997	201	0,971934	0,999036	0,999382	1,044107	0,980288	68,14069	76,12
302	347	0,004996	202	0,973523	1,000862	1,004654	1,074752	1,016801	69,49216	77,85
302	348	0,004997	202	0,975143	1,002729	1,010058	1,105306	1,054513	70,80546	79,58
302	349	0,004998	493	0,979824	1,008138	1,025835	1,190126	1,166436	74,25681	84,4
302	350	0,004998	533	0,98514	1,0143	1,044052	1,28124	1,299456	77,63305	89,7
302	452	0,004953	1	0,98514	1,0143	1,044052	1,28124	1,299456	77,87347	90
302	499	0,004997	149	0,986316	1,016243	1,048127	1,303431	1,333253	78,72296	91,6

302	499	0,004997	1	0,986316	1,016243	1,048127	1,303431	1,333253	78,72296	91,6
302	499	0,004999	1	0,987492	1,011285	1,05223	1,297298	1,338046	78,72015	91,6
303	300	-0,00499	1955	0,942684	0,978337	0,907535	-0,28355	0,191115	-1,87506	-5,61
303	301	-0,00229	1	0,942684	0,978337	0,907535	-0,28355	0,191115	-1,877	-3,59
303	302	0,000199	1	0,942684	0,978337	0,907535	-0,28355	0,191115	-1,87895	-1,73
303	303	0,002515	1	0,942684	0,978337	0,907535	-0,28355	0,191115	-1,8809	0
303	304	0,004751	1	0,942684	0,978337	0,907535	-0,28355	0,191115	-1,88284	1,67
303	305	0,004993	190	0,942232	0,977881	0,906181	-0,25331	0,198329	-0,39181	3,34
303	306	0,004993	207	0,941789	0,977435	0,904859	-0,21975	0,206655	1,277434	5,01
303	307	0,004992	204	0,941415	0,977055	0,903741	-0,18671	0,215191	2,934945	6,67
303	308	0,004992	203	0,941103	0,976738	0,90281	-0,15382	0,224042	4,599546	8,34
303	309	0,005	200	0,940854	0,976484	0,902069	-0,12141	0,233118	6,25307	10,007
303	310	0,004998	199	0,940664	0,97629	0,901505	-0,08917	0,242514	7,911253	11,674
303	311	0,005	197	0,940534	0,976156	0,901117	-0,05726	0,252183	9,564347	13,341
303	312	0,004996	196	0,940461	0,976081	0,900899	-0,02554	0,262176	11,21949	15,008
303	313	0,004997	194	0,940443	0,976063	0,900848	0,005835	0,272448	12,86759	16,675
303	314	0,004995	193	0,94048	0,976101	0,900958	0,037022	0,283055	14,51641	18,342
303	315	0,004999	191	0,94057	0,976195	0,901225	0,06787	0,293952	16,15711	20,009

303	316	0,004991	191	0,940712	0,976343	0,901648	0,098681	0,30525	17,80501	21,676
303	317	0,004993	189	0,940905	0,976544	0,90222	0,129129	0,316839	19,44201	23,343
303	318	0,004994	188	0,941146	0,976797	0,902939	0,159396	0,328791	21,07701	25,01
303	319	0,004995	187	0,941436	0,977103	0,903804	0,189477	0,341112	22,70899	26,677
303	320	0,004997	186	0,941773	0,977459	0,904809	0,219373	0,353809	24,3372	28,344
303	321	0,004991	186	0,942158	0,977866	0,90596	0,249221	0,366952	25,96839	30,011
303	322	0,004998	184	0,942586	0,978321	0,90724	0,27872	0,380415	27,58518	31,678
303	323	0,004998	184	0,94306	0,978827	0,908661	0,308197	0,394355	29,20471	33,345
303	324	0,004993	184	0,94358	0,979382	0,910221	0,33762	0,40877	30,82431	35,012
303	325	0,004994	183	0,944141	0,979984	0,911909	0,366833	0,423597	32,43456	36,679
303	326	0,004992	183	0,944747	0,980636	0,913733	0,396	0,438927	34,04352	38,346
303	327	0,004997	182	0,945392	0,981333	0,915682	0,424971	0,454694	35,64197	40,013
303	328	0,004999	182	0,946081	0,982078	0,917765	0,45391	0,470999	37,23806	41,68
303	329	0,004999	182	0,946811	0,982871	0,91998	0,482806	0,487853	38,83008	43,347
303	330	0,005	182	0,947583	0,983711	0,922326	0,511652	0,505267	40,4167	45,014
303	331	0,004992	183	0,948399	0,984603	0,924814	0,540585	0,523345	42,00423	46,681
303	332	0,004997	182	0,949251	0,985537	0,927418	0,5693	0,541912	43,57489	48,348
303	333	0,004994	183	0,950147	0,986521	0,930165	0,598119	0,561194	45,14512	50,015

303	334	0,004995	183	0,951082	0,987552	0,933039	0,626875	0,581103	46,70462	51,682
303	335	0,004999	183	0,952055	0,988628	0,936041	0,655583	0,601666	48,25301	53,349
303	336	0,004999	184	0,953072	0,989755	0,939187	0,684394	0,623016	49,79707	55,016
303	337	0,004996	185	0,954131	0,990934	0,942476	0,713287	0,645169	51,33426	56,683
303	338	0,004999	185	0,955227	0,992156	0,94589	0,742113	0,668035	52,85534	58,35
303	339	0,004999	186	0,956365	0,993429	0,94945	0,771035	0,69177	54,36744	60,017
303	340	0,004999	187	0,957545	0,994753	0,953154	0,800041	0,716398	55,86844	61,684
303	341	0,004998	188	0,958766	0,996127	0,957003	0,829126	0,741948	57,35646	63,351
303	342	0,004997	189	0,960028	0,997552	0,960997	0,858287	0,768454	58,82978	65,018
303	343	0,004996	190	0,96133	0,999026	0,965137	0,887523	0,79595	60,28672	66,685
303	344	0,004996	191	0,962673	1,000551	0,969423	0,916833	0,824474	61,72563	68,352
303	345	0,004996	192	0,964056	1,002125	0,973857	0,946219	0,854065	63,14489	70,019
303	346	0,004996	193	0,965478	1,003749	0,97844	0,975679	0,884764	64,5428	71,686
303	347	0,004996	194	0,96694	1,005423	0,98317	1,005212	0,91661	65,91765	73,353
303	348	0,004998	462	0,971119	1,010223	0,996826	1,086035	1,009554	69,5281	77,9
303	349	0,004997	492	0,975801	1,015615	1,012346	1,171223	1,117394	73,07842	82,7
303	350	0,004998	520	0,980975	1,021593	1,029783	1,2602	1,24191	76,47845	87,8
303	351	0,004996	1	0,980975	1,021593	1,029783	1,2602	1,24191	76,48107	87,8

303	451	0,004907	1	0,980975	1,021593	1,029783	1,2602	1,24191	76,73201	88
303	452	0,004992	1	0,980975	1,021593	1,029783	1,2602	1,24191	76,73441	88,2
303	453	0,004999	23	0,981147	1,021849	1,030366	1,263335	1,246462	76,84829	88,4
303	498	0,005	164	0,98245	1,023991	1,034807	1,287927	1,282494	77,80665	90
303	499	0,004999	180	0,983908	1,026391	1,039797	1,315055	1,323409	78,71945	91,6
303	500	0,004998	198	0,985543	1,02909	1,045424	1,345048	1,370114	79,69107	93,4

NOTE:

Untuk Penjelasan dan Data lebih lengkap
dapat menghubungi Penulis

LAMPIRAN B

(Listing Program of Modified ANFIS Algorithm)

```

/*****
* Program ini dibuat dengan menggunakan cross-
compiler Mbed OS dengan Keil U Vision 5
* Program ini dibuat bersama oleh Tim Solar
Tracker:
- Rezza MW : Program Modifier
- Ergi QN : Program Debugger
- Kevin KS : Program Analyst
- Hesty AWK : Program Designer
- Fiqiyah UA : Program Supplier

Semoga Bermanfaat :v :v
*****/
// Library
#include "mbed.h"
#include "rtos.h"
#include "SDFileSystem.h"

// Function Prototypes
void sdcard();
void getADC_LDR();
float positive(float p_);
float negative(float n_);
void goForward();
void goBackward();
void accelerate();
void decelerate();
void stop();
void pressed();

// Variables

```

```

float Arus, Tegangan, Arus_ADC, Tegangan_ADC,
del_v, Daya;
float LDR1, LDR2, LDR1_temp, LDR2_temp,
LDR1_aver, LDR2_aver, out_temp, m1, m2, out_f;
int checker1, checker2;
int direction = 0, dir1, dir2;
bool running = true;
int state = 1;
float stepFreq = 5;
int accelerationMode = 0;
int position = 1;

// Pheripherals
DigitalOut out1(PB_10);
DigitalOut out2(PB_4);
DigitalOut out3(PB_5);
DigitalOut out4(PB_3);
InterruptIn my_button(USER_BUTTON);
Thread thread;
DigitalOut led(LED1);
AnalogIn analog_Arus(PC_5);
AnalogIn analog_Tegangan (PC_4);
AnalogIn analog_LDR1(PC_2);
AnalogIn analog_LDR2(PC_3);
Serial pc(PA_11, PA_12);
SDFileSystem sd(D11, D12, D13, D10, "sd");

// OS Thread
void thread_AVP()
{
while (true) {
wait(0.3);
Tegangan_ADC = analog_Tegangan.read();
Arus_ADC = analog_Arus.read();
Tegangan = (((Tegangan_ADC/3.3)*4096)/100)+2.5;
del_v = (((Arus_ADC/3.3)*4096)/100)-2.5;

```

```

Arus = del_v/10 ;
pc.printf("V%.1f \n", Tegangan);
pc.printf("A%.1f\n", Arus);
Daya=Tegangan*Arus;
pc.printf("P%.1f \n", Daya);
sdcard();
}
}

void thread_NN(){
float x1, x2, x1_norm, x2_norm;
float Z_inj, Zj, voj, vij1, vij2;
float Y_ink, Yk, wok, wjk;
float out_denorm;
float in1, in2;

in1 = LDR1_aver;
in2 = LDR2_aver;

if(in1 <= 350) {
if(in1 <= 300) {
if(in2 <= 325) {
voj=0.93; vij1=0.95; vij2=0.95;
if(in2 <= 310) {
wok=0.24; wjk=-0.18;
}
else if(in2 > 310 && in2 < 320) {
wok=0.35; wjk=0.21;
}
else {
wok=0.48; wjk=0.42;
}
}
else if(in2 > 325 && in2 < 350) {
voj=0.97; vij1=0.97; vij2=0.97;
if(in2 < 340) {

```



```

wok=0.7; wjk=0.8;
}
else{
wok=1.2; wjk=1.1;
}
}
else {
voj=1.05; vij1=0.99; vij2=0.99; wok=1.3;
wjk=1.4;
}
}
else if(in1 > 300 && in1 <= 310) {
if(in2 <= 325) {
voj=0.84; vij1=0.92; vij2=1.04;
if(in2 <= 310) {
wok=0.13; wjk=-0.46;
}
else if(in2 > 310 && in2 < 320) {
wok=0.22; wjk=-0.06;
}
else
wok=0.28; wjk=0.16;
}
else if(in2 > 325 && in2 < 350) {
voj=0.92; vij1=0.92; vij2=1.03;
if(in2 < 340) {
wok=0.38; wjk=0.4;
}
else{
wok=0.8; wjk=1.1;
}
}
else {
voj=0.97; vij1=0.96; vij2=1.1; wok=1.3; wjk=1.4;
}
}
}

```

```

else if(in1 > 310 && in1 <= 330) {
  if(in2 <= 325) {
    voj=0.3; vij1=1.05; vij2=1.44;
    if(in2 <= 310) {
      wok=0.13; wjk=-0.46;
    }
    else if(in2 > 310 && in2 < 320) {
      wok=0.06; wjk=-0.85;
    }
    else
      wok=0.14; wjk=-0.21;
  }
  else if(in2 > 325 && in2 < 350) {
    voj=0.29; vij1=1.03; vij2=1.41;
    if(in2 < 340) {
      wok=0.4; wjk=-0.02;
    }
    else{
      wok=0.83; wjk=1.12;
    }
  }
  else {
    voj=0.37; vij1=1.1; vij2=1.5; wok=1.4; wjk=1.3;
  }
}
else if(in1 > 330 && in1 <= 350) {
  if(in2 <= 325) {
    if(in2 <= 310) {
      voj=0.03; vij1=0.4; vij2=2.35; wok=0.000038;
      wjk=-2.67;
    }
    else if(in2 > 310 && in2 < 320) {
      voj=0.02; vij1=0.14; vij2=2.25; wok=0.00025;
      wjk=-1.3;
    }
  }
  else

```

```
voj=0.018; vij1=0.95; vij2=2.2; wok=0.00044;
wjk=-0.9;
}
else if(in2 > 325 && in2 < 350) {
voj=0.016; vij1=0.057; vij2=2.16;
if(in2 < 340) {
wok=0.00078; wjk=-0.47;
}
else{
wok=0.002; wjk=0.08;
}
}
else {
voj=0.041; vij1=0.37; vij2=2.6; wok=0.086;
wjk=2.92;
}
}
}
else if (in1 > 351 && in1 < 500) {
if(in1 <= 449) {
if(in2 < 350) {
voj=0.0054; vij1=-0.31; vij2=3.2;
if(in2 <= 325) {
wok=0.000000226; wjk=-3.61;
}
else {
wok=0.00000352; wjk=-0.32;
}
}
}
else if(in2 >= 350 && in2 <= 449){
if(in2 <= 400) {
voj=0.0078; vij1=-0.32; vij2=3.1;
wok=0.000000726; wjk=-3.47;
}
}
else{
```

```

voj=0.001; vij1=-0.91; vij2=2.48; wok=0.0000063;
wjk=-0.56;
}
}
else {
if(in2 < 500) {
voj=0.002; vij1=-0.58; vij2=2.68; wok=0.00066;
wjk=2;
}
else {
voj=0.004; vij1=-0.42; vij2=2.92; wok=0.0018;
wjk=2.83;
}
}
}
else if(in1 >= 450 && in1 < 460) {
if(in2 < 350) {
voj=0.0016; vij1=-0.31; vij2=2.8;
if(in2 <= 325) {
wok=0.00000274; wjk=-2.74;
}
else {
wok=0.0000365; wjk=-1.06;
}
}
else if(in2 >= 350 && in2 <= 449){
if(in2 <= 400) {
voj=0.0037; vij1=-0.47; vij2=2.34;
wok=0.000000726; wjk=-3.47;
}
else{
voj=0.0035; vij1=-0.46; vij2=2.36; wok=0.00033;
wjk=0.46;
}
}
else {

```

```

if(in2 < 500) {
voj=0.0046; vij1=-0.38; vij2=2.48; wok=0.0013;
wjk=1.42;
}
else {
voj=0.0096; vij1=-0.113; vij2=2.85; wok=0.0086;
wjk=2.64;
}
}
}
else if(in1 >= 460 && in1 < 480) {
if(in2 < 350) {
voj=0.028; vij1=-0.29; vij2=2.13;
if(in2 <= 325) {
wok=0.000000026; wjk=-3.61;
}
else {
wok=0.000000099; wjk=-3.06;
}
}
else if(in2 >= 350 && in2 <= 449){
if(in2 <= 400) {
voj=0.0063; vij1=-1.17; vij2=2.51;
wok=0.0000013; wjk=-0.77;
}
else{
voj=0.0065; vij1=-1.08; vij2=2.46; wok=0.000053;
wjk=0.13;
}
}
else {
if(in2 < 500) {
voj=0.0069; vij1=-1.18; vij2=2.55;
wok=0.0000089; wjk=0.29;
}
else {

```

```

voj=0.0087; vij1=-0.97; vij2=2.64; wok=0.000046;
wjk=1.64;
}
}
}
else {
if(in2 < 350) {
if(in2 <= 325) {
voj=0.0052; vij1=-0.32; vij2=3.2;
wok=0.000000032; wjk=-3.35;
}
else {
voj=0.0011; vij1=-0.86; vij2=2.71;
wok=0.0000003; wjk=-1.8;
}
}
else if(in2 >= 350 && in2 <= 449){
if(in2 <= 400) {
voj=0.00063; vij1=-1.06; vij2=2.41;
wok=0.000014; wjk=0.64;
}
else{
voj=0.0008; vij1=-0.98; vij2=2.6; wok=0.000041;
wjk=1.35;
}
}
else {
if(in2 < 500) {
voj=0.00128; vij1=-0.82; vij2=2.85; wok=0.00015;
wjk=2.3;
}
else {
voj=0.0022; vij1=-0.64; vij2=3.17; wok=0.00039;
wjk=3.21;
}
}
}

```

```

}
}
else if(in1 >= 501 && in1 <=550){
if(in1 <= 520) {
if(in2 <= 300) {
voj=0.58; vij1=1.37; vij2=1.75; wok=0.0036;
wjk=-2.96;
}
else if(in2 > 300 && in2 < 449) {
voj=0.0022; vij1=1.34; vij2=1.75; wok=0.0036;
wjk=-2.96;
}
else if(in2 >= 450 && in2 <= 475) {
voj=0.0022; vij1=1.25; vij2=1.5; wok=0.00012;
wjk=-2.2;
}
else if(in2 > 475 && in2 <= 500) {
voj=0.41; vij1=1.17; vij2=1.42; wok=0.001; wjk=-
0.76;
}
else if(in2 > 500 && in2 <= 525) {
voj=0.4; vij1=1.16; vij2=1.4; wok=0.004; wjk=-
0.47;
}
else {
voj=0.29; vij1=1; vij2=1.7; wok=0.00012;
wjk=2.7;
}
}
else if(in1 > 520 && in1 < 540) {
if(in2 <= 300) {
voj=0.0058; vij1=-0.92; vij2=4.3;
wok=0.000000000000000000000000000067; wjk=-3.68;
}
else if(in2 > 300 && in2 < 449) {

```



```

voj=0.000024; vij1=-3.55; vij2=9.54; wok=0;
wjk=-1.77;
}
else if(in2 > 525 && in2 <= 550) {
voj=0.00049; vij1=-4.11; vij2=9.62; wok=0; wjk=-
0.34;
}
else if(in2 > 550 && in2 <= 675) {
voj=0.0004; vij1=-4.35; vij2=10.23; wok=0; wjk=-
0.98;
}
else {
voj=0.0000018; vij1=-4.48; vij2=10.92; wok=0;
wjk=1.32;
}
}
}
else if(in1 > 550) {
if(in1 <= 650) {
if(in2 <= 520) {
voj=0.0038; vij1=-0.255; vij2=3.28;
wok=0.0000000000437; wjk=-3.64;
}
else if(in2 > 520 && in2 <= 540) {
voj=0.000824; vij1=-1.25; vij2=2.46;
wok=0.0000000014; wjk=-1.15;
}
else if(in2 > 540 && in2 <= 650) {
voj=0.00066; vij1=-1.41; vij2=2.36;
wok=0.00000000431; wjk=-0.47;
}
else if(in2 > 650 && in2 <= 670) {
voj=0.000654; vij1=-1.4; vij2=2.34;
wok=0.0000000184; wjk=0.42;
}
else if(in2 > 670 && in2 <= 690) {

```

```

voj=0.000938; vij1=-1.18; vij2=2.59;
wok=0.000000101; wjk=1.59;
}
else if(in2 > 690){
voj=0.00165; vij1=-0.811; vij2=2.98;
wok=0.000000468; wjk=2.79;
}
}
else if(in1 > 650 && in1 <= 665) {
if(in2 <= 525) {
voj=0.00148; vij1=0.653; vij2=2.92; wok=0; wjk=-
3.2;
}
else if(in2 > 525 && in2 <= 540) {
voj=0.0086; vij1=0.29; vij2=2.63; wok=0; wjk=-
1.62;
}
else if(in2 > 540 && in2 <= 650) {
voj=0.0073; vij1=0.184; vij2=2.54; wok=0; wjk=-
0.87;
}
else if(in2 > 650 && in2 <= 670) {
voj=0.0069; vij1=0.153; vij2=2.52; wok=0; wjk=-
0.42;
}
else if(in2 > 670 && in2 <= 690) {
voj=0.0072; vij1=0.173; vij2=2.54; wok=0;
wjk=0.79;
}
else if(in2 > 690){
voj=0.0086; vij1=0.25; vij2=2.62; wok=0;
wjk=1.59;
}
}
else if(in1 > 665 && in1 <= 680) {
if(in2 <= 540) {

```

```

voj=0.17; vij1=2.29; vij2=2.69; wok=0; wjk=-
2.73;
}
else if(in2 > 540 && in2 <= 650) {
voj=0.154; vij1=2.22; vij2=2.64; wok=0; wjk=-
1.67;
}
else if(in2 > 650 && in2 <= 670) {
voj=0.148; vij1=2.19; vij2=2.61; wok=0; wjk=-
0.54;
}
else if(in2 > 670 && in2 <= 690) {
voj=0.147; vij1=2.194; vij2=2.61; wok=0;
wjk=0.232;
}
else if(in2 > 690){
voj=0.149; vij1=2.2; vij2=2.61; wok=0; wjk=0.85;
}
}
else if(in1 > 680 && in1 <= 695) {
if(in2 <= 650) {
voj=0.52; vij1=3.05; vij2=2.77; wok=0; wjk=-
2.61;
}
else if(in2 > 650 && in2 <= 670) {
voj=0.5; vij1=3.04; vij2=2.76; wok=0; wjk=-0.88;
}
else if(in2 > 670 && in2 <= 690) {
voj=0.5; vij1=3.03; vij2=2.75; wok=0; wjk=-0.26;
}
else if(in2 > 690){
voj=0.5; vij1=3.03; vij2=2.75; wok=0; wjk=0.19;
}
}
else if(in1 > 695) {
if(in2 <= 665) {

```

```

voj=0.78; vij1=3.34; vij2=2.81; wok=0; wjk=-
2.58;
}
else if(in2 > 665 && in2 <= 680) {
voj=0.76; vij1=3.32; vij2=2.8; wok=0; wjk=-0.99;
}
else if(in2 > 680 && in2 <= 690) {
voj=0.76; vij1=3.32; vij2=2.79; wok=0; wjk=-
0.69;
}
else if(in2 > 690){
voj=0.76; vij1=3.32; vij2=2.79; wok=0; wjk=-
0.12;
}
}
}
}

```

```

x1 = LDR1_aver;
x2 = LDR2_aver;

```

```

x1_norm = (((x1-0)/(1000-0))*((1-0)+(0)));
x2_norm = (((x2-0)/(1000-0))*((1-0)+(0)));

```

```

Z_inj = voj + ((x1_norm*vij1)+(x2_norm*vij2));
Zj = 1/(1+(exp(-Z_inj)));

```

```

Y_ink = wok + ((Zj*wjk)+(Zj*wjk));
Yk = 1/(1+(exp(-Y_ink)));

```

```

out_temp = Yk;
}

```

```

void thread_infrule(){
float in1, in2;
float neg, pos, min, plus;

```

```
in1 = LDR1_aver;  
in2 = LDR2_aver;  
out_f = out_temp;
```

```
if(in1 <= 350) {  
  checker1 = 1;  
}  
if(in1 >= 300 && in1 <=500) {  
  checker1 = 2;  
}  
if(in1 >= 450 && in1 <= 550) {  
  checker1 = 3;  
}  
if(in1 >= 500 && in1 <= 700) {  
  checker1 = 4;  
}  
if(in1 >= 650) {  
  checker1 = 5;  
}  
if(in2 <= 350) {  
  checker2 = 1;  
}  
if(in2 >= 300 && in2 <=500) {  
  checker2 = 2;  
}  
if(in2 >= 450 && in2 <= 550) {  
  checker2 = 3;  
}  
if(in2 >= 500 && in2 <= 700) {  
  checker2 = 4;  
}  
if(in2 >= 650) {  
  checker2 = 5;  
}
```

```
if (checker1 == checker2) {
    out_f = 0;
    m1 = out_f;
    m2 = out_f;
}
else if(checker1 > checker2) {
    dir1 = 0;
    dir2 = 1;
    stepFreq = stepFreq + 1;
    if (out_f < 0) {
        min = out_f;
        plus = out_f;
        m1 = min;
        m2 = positive(plus);
    }
    else if (out_f > 0) {
        min = out_temp;
        plus = out_temp;
        m1 = negative(min);
        m2 = plus;
    }
}
else if(checker1 < checker2) {
    dir1 = 1;
    dir2 = 0;
    stepFreq = stepFreq - 1;
    if (out_f > 0) {
        plus = out_f;
        min = out_f;
        m1 = plus;
        m2 = negative(min);
    }
    else if (out_f < 0) {
        plus = out_f;
        min = out_f;
        m1 = positive(plus);
    }
}
```

```

m2 = min;
}
}
}

```

```

void thread_defuzifikasi() {
float m1_denorm, m2_denorm;

```

```

m1_denorm = ((m1-(0))/(1-(0)))*(93,43-(-
93,43))+(-93,43);
m2_denorm = ((m2-(0))/(1-(0)))*(93,43-(-
93,43))+(-93,43);
}

```

```

// Main Program
int main()
{
int pil;

```

```

pc.readable();
pc.baud(57600);

```

```

thread.start(thread_AVP);
my_button.fall(&pressed);

```

```

while (true) {
getADC_LDR();
wait(0.3);
led = 1;
if(running == true) {
halfStep(position);
position = increment(position , dir2, out_f);
led = !led;
}
wait(1/stepFreq);
}

```

```

}

// Functions
void sdcard()
{
    printf("SD thread is ready\r\n");
    mkdir("/sd/Progress2", 0777);

    FILE *fp = fopen("/sd/Progress2/sdtest.txt",
        "a+");
    if(fp == NULL) {
        error("Could not open file for write\n");
    }
    fprintf(fp, "Arus=%f\t Tegangan=%f\t
    Daya=%f\t\n", Arus ,Tegangan, Daya);
    fclose(fp);
}

void getADC_LDR()
{
    wait(0.3);
    for(int a=0; a<77; a++) {
        LDR1 = analog_LDR1.read();
        LDR2 = analog_LDR2.read();
        LDR1_temp = LDR1_temp + LDR1;
        LDR2_temp = LDR2_temp + LDR2;
    }

    LDR1_aver = LDR1_temp / 77;
    LDR2_aver = LDR2_temp / 77;
    pc.printf("x%.2f \n", LDR1_aver);
    pc.printf("y%.2f\n", LDR2_aver);
    wait(0.5);
    thread.start(thread_NN);
    wait(0.2);
    thread.start(thread_infrule);
}

```



```
wait(0.2);  
thread.start(thread_defuzifikasi);  
}
```

```
void halfStep(int position) {  
    switch(position){  
        case 1 :  
            out1.write(0);  
            out2.write(1);  
            out3.write(1);  
            out4.write(0);  
            break;  
        case 2 :  
            out1.write(1);  
            out2.write(0);  
            out3.write(0);  
            out4.write(1);  
            break;  
        case 3 :  
            out1.write(0);  
            out2.write(1);  
            out3.write(0);  
            out4.write(0);  
            break;  
        case 4 :  
            out1.write(0);  
            out2.write(1);  
            out3.write(1);  
            out4.write(0);  
            break;  
        case 5 :  
            out1.write(0);  
            out2.write(0);  
            out3.write(1);  
            out4.write(0);  
            break;  
    }
```

```
case 6 :
out1.write(0);
out2.write(0);
out3.write(1);
out4.write(1);
break;
case 7 :
out1.write(0);
out2.write(0);
out3.write(0);
out4.write(1);
break;
case 8 :
out1.write(1);
out2.write(0);
out3.write(0);
out4.write(1);
break;
}
}
```

```
int increment(int position ,int direction, int
upperLimit) {
if(direction == 0) {
if(position > upperLimit - 1) {
position = 1;
}
else {
position++;
}
}
else {
if(position < 2) {
position = upperLimit;
}
else {
```

```
position--;  
}  
}  
return position;  
}
```

```
void pressed() {  
if(state < 3) {  
state++;  
}  
else {  
state = 1;  
}  
switch(state){  
case 1 :  
stop();  
break;  
case 2 :  
goBackward();  
break;  
case 3 :  
goForward();  
break;  
}  
}
```

```
void goForward() {  
accelerationMode = 0;  
running = true;  
direction = 0;  
}
```

```
void goBackward() {  
accelerationMode = 0;  
running = true;  
direction = 1;
```

```
}
```

```
void stop() {  
    running = false;  
}
```

```
/******
```

```
Selesai ~~
```

```
Atau Tidak Terserah pada Developer
```

```
Semoga Bermanfaat :v :v
```

```
*****/
```


LAMPIRAN C

(Datasheet Nucleo STM32F401RE)



UM1724

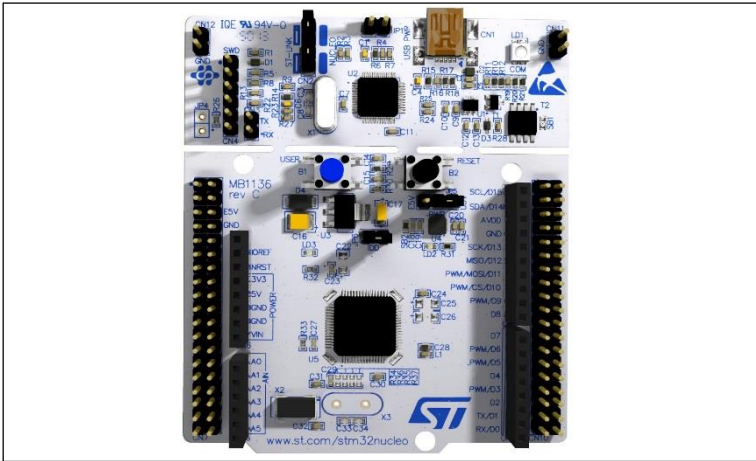
User manual

STM32 Nucleo-64 boards

Introduction

The STM32 Nucleo-64 boards (NUCLEO-F030R8, NUCLEO-F070RB, NUCLEO-F072RB, NUCLEO-F091RC, NUCLEO-F103RB, NUCLEO-F302R8, NUCLEO-F303RE, NUCLEO-F334R8, NUCLEO-F401RE, NUCLEO-F410RB, NUCLEO-F411RE, NUCLEO-F446RE, NUCLEO-L010RB, NUCLEO-L053R8, NUCLEO-L073RZ, NUCLEO-L152RE, NUCLEO-L452RE, NUCLEO-L476RG) provide an affordable and flexible way for users to try out new concepts and build prototypes with the STM32 microcontrollers in LQFP64 package, choosing from the various combinations of performance, power consumption and features. The Arduino[™] Uno V3 connectivity support and the ST morpho headers allow to expand easily the functionality of the Nucleo open development platform with a wide choice of specialized shields. The STM32 Nucleo boards do not require any separate probe as they integrate the ST-LINK/V2-1 debugger and programmer. The STM32 Nucleo boards come with the STM32 comprehensive software HAL library together with various packaged software examples, as well as direct access to the Arm[®] Mbed[™] online resources at <http://mbed.org/>.

Figure 1. STM32 Nucleo-64 board



Picture is not contractual.

1 Features

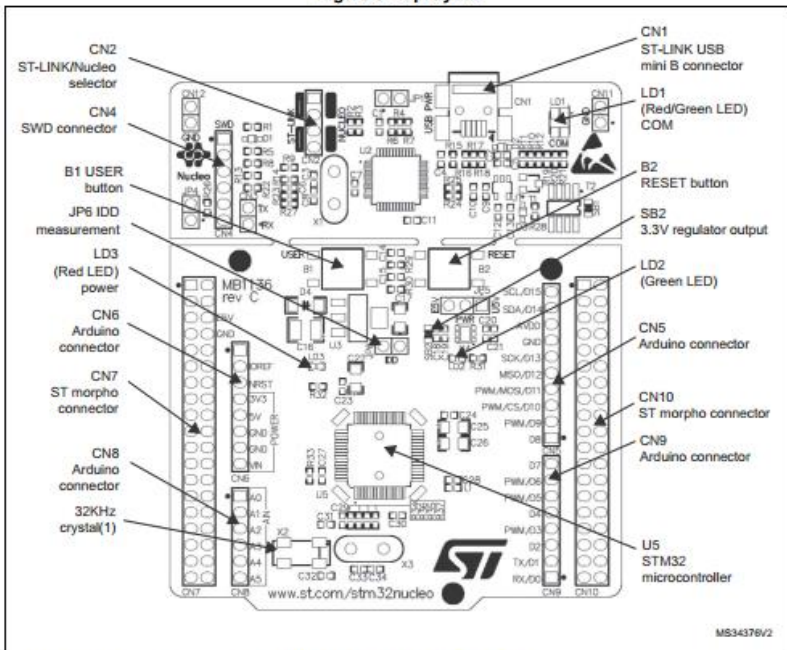
The STM32 Nucleo board offers the following features:

- STM32 microcontroller in LQFP64 package
- Three LEDs:
 - USB communication (LD1), user LED (LD2), power LED (LD3)
- Two push-buttons: USER and RESET
- Two types of extension resources
 - Arduino[™] Uno V3 connectivity
 - ST morpho extension pin headers for full access to all STM32 I/Os
- Flexible board power supply:
 - USB VBUS or external source (3.3 V, 5 V, 7 - 12 V)
 - Power management access point
- On-board ST-LINK/V2-1 debugger and programmer with SWD connector
- Selection-mode switch to use the kit as a standalone ST-LINK/V2-1

- USB re-enumeration capability. Three different interfaces supported on USB:
 - Virtual COM port
 - Mass storage
 - Debug port
- Comprehensive free software HAL library including a variety of software examples
- Arm[®] Mbed[™] (see <http://mbed.org>)

2 HARDWARE LAYOUT AND CONFIGURATION

Figure 3. Top layout



1. Crystal may be present or not depending on board version, refer to [Section 6.7.2](#).

Figure 4. Bottom layout

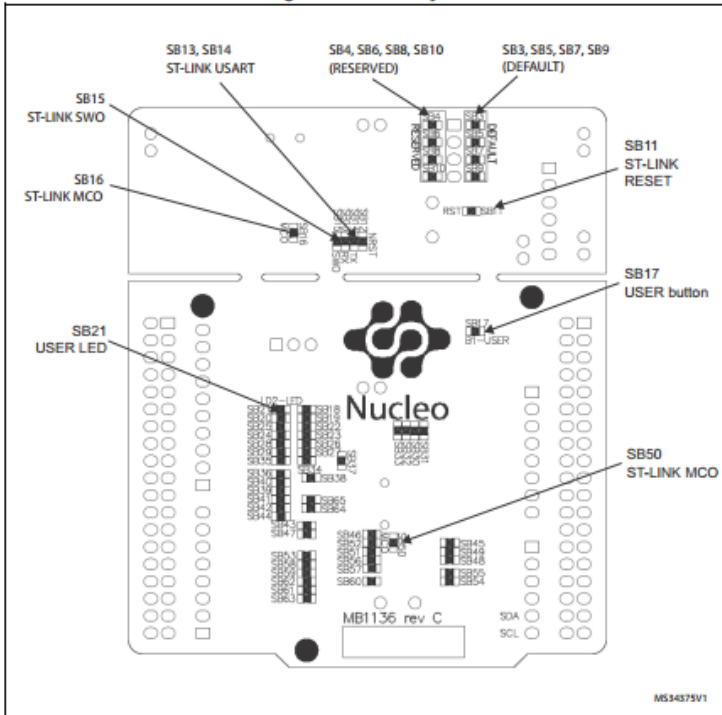


Figure 18. NUCLEO-F401RE

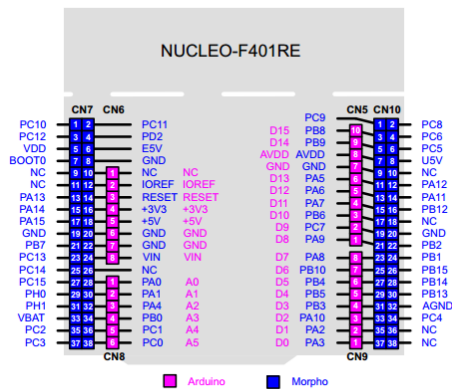


Table 16. Arduino connectors on NUCLEO-F401RE and NUCLEO-F411RE

Connector	Pin	Pin name	STM32 pin	Function
Left connectors				
CN6 power	1	NC	-	-
	2	IOREF	-	3.3V Ref
	3	RESET	NRST	RESET
	4	+3.3V	-	3.3V input/output
	5	+5V	-	5V output
	6	GND	-	ground
	7	GND	-	ground
	8	VIN	-	Power input
CN8 analog	1	A0	PA0	ADC1_0
	2	A1	PA1	ADC1_1
	3	A2	PA4	ADC1_4
	4	A3	PB0	ADC1_8
	5	A4	PC1 or PB9 ⁽¹⁾	ADC1_11 (PC1) or I2C1_SDA (PB9)
	6	A5	PC0 or PB8 ⁽¹⁾	ADC1_10 (PC0) or I2C1_SCL (PB8)
Right connectors				
CN5 digital	10	D15	PB8	I2C1_SCL
	10	D15	PB8	I2C1_SCL
	9	D14	PB9	I2C1_SDA
	8	AREF	-	AVDD
	7	GND	-	ground

Table 16. Arduino connectors on NUCLEO-F401RE and NUCLEO-F411RE (continued)

Connector	Pin	Pin name	STM32 pin	Function
CN5 digital	6	D13	PA5	SPI1_SCK
	5	D12	PA6	SPI1_MISO
	4	D11	PA7	TIM1_CH1N or SPI1_MOSI
	3	D10	PB6	TIM4_CH1 or SPI1_CS
	2	D9	PC7	TIM3_CH2
	1	D8	PA9	-
CN9 digital	8	D7	PA8	-
	7	D6	PB10	TIM2_CH3
	6	D5	PB4	TIM3_CH1
	5	D4	PB5	-
	4	D3	PB3	TIM2_CH2
	3	D2	PA10	-
	2	D1	PA2	USART2_TX
	1	D0	PA3	USART2_RX

**Table 29. ST morpho connector on NUCLEO-F401RE,
NUCLEO-F411RE, NUCLEO-F446RE**

CN7 odd pins		CN7 even pins		CN10 odd pins		CN10 even pins	
Pin	Name	Name	Pin	Pin	Name	Name	Pin
1	PC10	PC11	2	1	PC9	PC8	2
3	PC12	PD2	4	3	PB8	PC6	4
5	VDD	E5V	6	5	PB9	PC5	6
7	BOOT0 ⁽¹⁾	GND	8	7	AVDD	U5V ⁽²⁾	8
9	-	-	10	9	GND	-	10
11	-	IOREF	12	11	PA5	PA12	12
13	PA13 ⁽³⁾	RESET	14	13	PA6	PA11	14
15	PA14 ⁽³⁾	+3.3V	16	15	PA7	PB12	16
17	PA15	+5V	18	17	PB6	-	18
19	GND	GND	20	19	PC7	GND	20
21	PB7	GND	22	21	PA9	PB2	22
23	PC13	VIN	24	23	PA8	PB1	24
25	PC14	-	26	25	PB10	PB15	26
27	PC15	PA0	28	27	PB4	PB14	28
29	PH0	PA1	30	29	PB5	PB13	30
31	PH1	PA4	32	31	PB3	AGND	32
33	VBAT	PB0	34	33	PA10	PC4	34
35	PC2	PC1 or PB9 ⁽⁴⁾	36	35	PA2	-	36
37	PC3	PC0 or PB8 ⁽⁴⁾	38	37	PA3	-	38

For more see:

https://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf

LAMPIRAN D

(Datasheet Sensor LDR)



Email: info@sunrom.com or sunrom@gmail.com

Visit us at <http://www.sunrom.com>

Document: **Datasheet** Date: **28-Jul-08** Model #: **3190** Product's Page: www.sunrom.com/ko-510.html

Light Dependent Resistor - LDR

Two cadmium sulphide(cds) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.



Applications

Photoconductive cells are used in many different types of circuits and applications.

Analog Applications

- Camera Exposure Control
- Auto Slide Focus - dual cell
- Photocopy Machines - density of toner
- Colorimetric Test Equipment
- Densitometer
- Electronic Scales - dual cell
- Automatic Gain Control – modulated light source
- Automated Rear View Mirror

Digital Applications

- Automatic Headlight Dimmer
- Night Light Control
- Oil Burner Flame Out
- Street Light Control
- Absence / Presence (beam breaker)
- Position Sensor

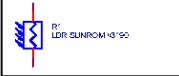
Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Unit
Cell resistance	1000 LUX	-	400	-	Ohm
	10 LUX	-	9	-	K Ohm
Dark Resistance	-	-	1	-	M Ohm
Dark Capacitance	-	-	3.5	-	pF
Rise Time	1000 LUX	-	2.8	-	ms
	10 LUX	-	18	-	ms
Fall Time	1000 LUX	-	48	-	ms
	10 LUX	-	120	-	ms
Voltage AC/DC Peak	-	-	-	320	V max
Current	-	-	-	75	mA max
Power Dissipation	-	-	-	100	mW max
Operating Temperature	-	-60	-	+75	Deg. C

Guide to source illuminations

Light source Illumination	LUX
Moonlight	0.1
60W Bulb at 1m	50
1W MES Bulb at 0.1m	100
Fluorescent Lighting	500
Bright Sunlight	30,000

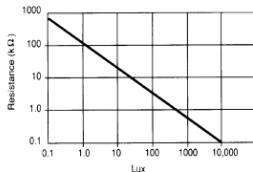
FIGURE 1 CIRCUIT SYMBOL



Sensitivity

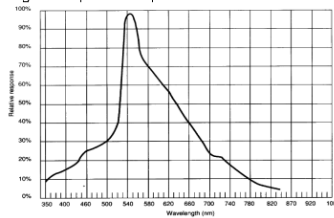
The sensitivity of a photodetector is the relationship between the light falling on the device and the resulting output signal. In the case of a photocell, one is dealing with the relationship between the incident light and the corresponding resistance of the cell.

FIGURE 2 RESISTANCE AS FUNCTION OF ILLUMINATION



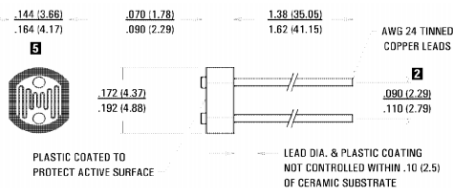
Spectral Response

Figure 3 Spectral response



Like the human eye, the relative sensitivity of a photoconductive cell is dependent on the wavelength (color) of the incident light. Each photoconductor material type has its own unique spectral response curve or plot of the relative response of the photocell versus wavelength of light.

Dimensions



Typical Application Circuits

Figure 6 Sensitive light operated relay

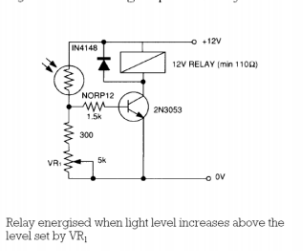


Figure 9 Logarithmic law photographic light meter

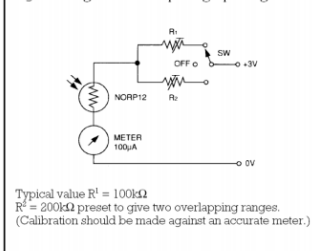
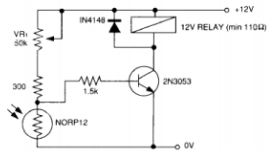


Figure 7 Light interruption detector



As Figure 6 relay energised when light level drops below the level set by VR₁

Figure 8 Automatic light circuit

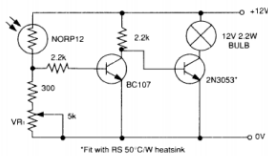
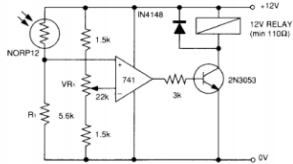


Figure 10 Extremely sensitive light operated relay



(Relay energised when light exceeds preset level.)
Incorporates a balancing bridge and op-amp. R₁ and NORP12 may be interchanged for the reverse function.

BIODATA PENULIS



Penulis yang bernama Rezza Maulana Wahid dilahirkan di Kabupaten Mojokerto pada tanggal 13 Juli 1997. Penulis merupakan anak pertama dari dua bersaudara, dari orang tua Bapak Wahid dan Ibu Nunuk. Penulis menempuh pendidikan formal SMAN 1 Gedeg Kemudian penulis melanjutkan ke jenjang yang lebih tinggi yaitu di Institut Teknologi Sepuluh Nopember Surabaya, tepatnya di Departemen Teknik Instrumentasi Fakultas Vokasi di tahun 2015 dan terdaftar dengan NRP 10 51 15 000 00 048.

Di Departemen Teknik Instrumentasi ini penulis aktif menjadi asisten praktikum untuk beberapa mata kuliah seperti Elektronika Industri, Sistem Pengukuran Listrik, Pemrograman Komputer, dan Mikroprosesor dan Interfacing. Pada laporan ini, penulis telah melaksanakan Tugas Akhir Program Studi DIII Teknologi Instrumentasi Departemen Teknik Instrumentasi FV ITS.

Selesaiannya Tugas Akhir ini diajukan penulis untuk memenuhi syarat memperoleh gelar Ahli Madya di Departemen Teknik Instrumentasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya.

Wahidr60@gmail.com